Working Implementation Agreements for Open Systems Interconnection Protocols: Part 27 - Open Document Architecture Level 2 to Level 3 Migration DAP

Output from the March 1994 Open Systems Environment Implementors' Workshop (OIW)

Acting SIG Chair: **Jon Stewart, Quality One Software Corporation**
SIG Editor: **Jim Wing, IBM Software Solutions Div.**

Foreword

This part of the Working Implementation Agreements was prepared by the of the Open Systems Environment Implementors' Workshop (OIW).

This text was approved by the Plenary of the Workshop.

This Specification is intended to be used *only* as a working agreement to aid the development of an Implementation Support Requirements (ISR) on the FOD36 | PM-36 DAP for the North American region. This ISR will eventually be added as the second part of Part 16 of the OIW Agreements, which currently contains only the FOD36 | PM-36 DAP.

This Specification is *not* intended to be a new Document Application Profile for the North American or a ny other region. Therefore this Specification uses the same DAP identifier as the FOD36 | PM-36 DAP.

Annex A forms an integral part of this part of the Specification. Annexes B and C are for information only.

—

Future changes and additions to this version of these Implementor Agreements will be published as a n ew part with redline (shaded) for next text and stikeout (---) for deleted text.

This part uses a convention of double and single quotes that has been established by ISO for use in th e ODA base standard and related document application profiles. The convention is to use within the tex t double quotes to accentuate ODA attribute names and single quotes to accentuate values for those attributes.

Contents

List of figures xi

List of tables xii

Part 27 - Open Document Architecture Level 2 to Level 3 Migration DAP 1

Introduction 1

1 Scope 2

2 Normative references 3
2.1 Paired Recommendations | International Standards equivalent in technical content 3
2.2 Additional references 5

3 Definitions 6

4 Relationship with other profiles 7

5 Conformance 7
5.1 Data stream conformance 7
5.2 Implementation conformance 8

6 Characteristics supported by this document application profile 9
6.1 Overview 9
6.1.1 General 9
6.1.2 Processable form documents 10
6.1.3 Formatted processable form documents 10
6.1.4 Generic documents 10
6.1.5 Document profile 11
6.2 Logical characteristics 11
6.2.1 Introduction 12
6.2.2 Overview of the logical structure 12
6.2.3 Body part of the logical structure 12
6.2.3.1 DocumentLogicalRoot 12
6.2.3.2 Passage 12
6.2.3.3 NumberedSegment 13
6.2.3.4 Number
. 14
6.2.3.5 Paragraph 15
6.2.3.6 BodyText, BodyRaster and BodyGeometric 15
6.2.3.7 Footnote 16
6.2.3.8 FootnoteReference 17
6.2.3.9 FootnoteBody 17

6.2.3.10 FootnoteNumber 17
6.2.3.11 FootnoteText 18
6.2.3.12 Phrase 18
6.2.3.12.1 Reference 19
6.2.3.12.2 ReferencedContent 19
6.2.3.13 Constituents used to represent tables 20
6.2.3.13.1 Introduction 20
6.2.3.13.2 Table 20
6.2.3.13.3 Row 20
6.2.3.13.4 EntryElement 20
6.2.3.13.5 EntryText, EntryRaster and EntryGeometric 20
6.2.3.14 Constituents representing lists 21
6.2.3.14.1 Introduction 21
6.2.3.14.2 NumberedList 21
6.2.3.14.3 UnNumberedList 21
6.2.3.14.4 DefinitionList 22
6.2.3.14.5 ListItem 22
6.2.3.14.6 ListTerm 22
6.2.4 Common content part of the logical structure 23
6.2.4.1 CommonContent 23
6.2.4.2 CommonText 23
6.2.4.3 PageNumber 23
6.2.4.4 CommonRaster 24
6.2.4.5 CommonGeometric 24
6.2.4.6 CommonReference 24
6.2.4.7 TableNumber 24
6.2.1 Overview of the layout characteristics 26
6.2.2 DocumentLayoutRoot 26
6.2.3 PageSet 27
6.2.4 Page characteristics 27
6.2.4.1 Page constituents 27
6.2.4.2 Page dimensions 28
6.2.4.3 Nominal page sizes 28
6.2.4.4 Page offset 29
6.2.4.5 Page layout characteristics 29
6.2.4.5.1 General characteristics 29
6.2.4.5.2 Body area layout 30
6.2.4.5.3 Header/footer area layout 30
6.2.4.5.4 Page layout 30
6.2.5 Body area characteristics 31
6.2.5.1 General characteristics 31
6.2.5.2 BasicBody 33
6.2.5.3 VariableCompositeBody 33
6.2.5.4 BasicFloat 34
6.2.5.5 SnakingColumns 35
6.2.5.6 SynchronizedColumns 36
6.2.5.7 ColumnVariable 36
6.2.5.8 ColumnFixed 37
6.2.5.9 FootnoteArea 38

6.2.5.10 Constituents used for laying out tables 38
6.2.5.10.1 Overview 38
6.2.5.10.2 TableArea 39
6.2.5.10.3 TableHeader 40
6.2.5.10.4 TableLabel 40
6.2.5.10.5 TableLabelContent 41
6.2.5.10.6 RowArea 41
6.2.5.10.7 Cell 41
6.2.6 Header and footer area characteristics 41
6.2.6.1 General characteristics 41
6.2.6.2 BasicHeader and BasicFooter 42
6.2.6.3 CompositeHeader and CompositeFooter 42
6.2.6.4 SourcedContentVariable 42
6.2.6.5 ArrangedContentVariable 43
6.2.6.6 SourcedContentFixed 43
6.2.6.7 ArrangedContentFixed 44
6.2.7 GenericBlock and SpecificBlock 44
6.3.1 Flow controls 45
6.3.1.1 Allocation of content to page sets 45
6.3.1.2 Page breaks 46
6.3.1.3 Allocation of content to body areas 46
6.3.1.3.1 General mechanism for laying out content in a composite body area 47
6.3.1.3.2 Layout into BasicFloat frames 48
6.3.1.3.3 Layout in SnakingColumns frames 48
6.3.1.3.4 Layout in SynchronizedColumns frames 48
6.3.1.3.5 Layout of tables 48
6.3.1.3.6 Layout of footnotes 49
6.3.1.4 Allocation of content to header and footer areas 50
6.3.2 Layout of document content 50
6.3.2.1 Margins 50
6.3.2.2 Separation 51
6.3.2.3 Indivisibility 52
6.3.2.4 Same layout object 52
6.3.2.5 Concatenation 52
6.3.2.6 Block alignment 52
6.3.3 Layout controls applicable in the absence of a generic layout structure 52
6.4 Content layout and imaging characteristics 53
6.4.1 Character content 53
6.4.1.1 Introduction 53
6.4.1.2 Character content architecture class 55
6.4.1.3 Character repertoires 55
6.4.1.4 Code extension techniques 56
6.4.1.5 Line spacing 57
6.4.1.6 Character spacing 57
6.4.1.7 Character path and line progression 58
6.4.1.8 Character positioning controls 59
6.4.1.9 Character orientation 59
6.4.1.10 Emphasis 59
6.4.1.11 Tabulation 60

6.4.1.12 Alignment . . . 61
6.4.1.13 First line format . . . 61
6.4.1.14 Widow and orphan sizes . . . 62
6.4.1.15 Fonts . . . 62
6.4.1.16 Kerning offset . . . 63
6.4.1.17 Proportional line spacing . . . 63
6.4.1.18 Superscripts and subscripts . . . 63
6.4.1.19 Line breaks . . . 63
6.4.1.20 Substitution of characters . . . 63
6.4.1.21 Initial point . . . 63
6.4.1.22 Use of control functions . . . 63
6.4.1.23 Formatting the content . . . 64
6.4.2 Raster graphics content . . . 65
6.4.2.1 Introduction . . . 65
6.4.2.2 Raster graphics content architecture . . . 65
6.4.2.3 Raster graphics encoding methods . . . 65
6.4.2.4 Pel path and line progression . . . 67
6.4.2.5 Clipping . . . 67
6.4.2.6 Pel spacing . . . 68
6.4.2.7 Spacing ratio . . . 68
6.4.2.8 Image dimensions . . . 68
6.4.3 Geometric graphics content . . . 68
6.5.1 Resource-documents . . . 70
6.5.2 External-documents . . . 70
6.5.3 Border . . . 70
6.5.4 Application comments . . . 70
6.5.5 Alternative representation . . . 71
6.5.6 Automatic numbering and referencing mechanisms . . . 71
6.5.6.1 Introduction . . . 71
6.5.6.2 Bindings . . . 71
6.5.6.2.1 Binding 'prefixes-<n>' . . . 72
6.5.6.2.2 Binding 'suffixes-<n>' . . . 72
6.5.6.2.3 Binding 'numberstring-<n>' . . . 72
6.5.6.2.4 Binding 'numbers-<n>' . . . 72
6.5.6.2.5 Binding 'fnotestring' . . . 72
6.5.6.3 Numbering of segments . . . 73
6.5.6.4 Numbering of lists . . . 74
6.5.6.5 Numbering of tables . . . 74
6.5.6.6 Footnote numbering . . . 75
6.5.6.7 Page numbering . . . 75
6.5.6.8 Common references . . . 77
6.5.7 User readable comments . . . 77
6.5.8 User visible name . . . 78
6.6 Document management features . . . 78
6.6.1 Document constituent information . . . 78
6.6.1.1 Presence of document constituents . . . 78
6.6.1.2 Resource-document information . . . 79
6.6.1.3 External-document information . . . 79
6.6.2 Document identification information . . . 79

6.6.2.1 Document application profile information 79
6.6.2.2 Document architecture class information 79
6.6.2.3 Content architecture classes information 79
6.6.2.4 ODA version information 79
6.6.2.5 Document reference 80
6.6.3 Document default information 80
6.6.4 Non-basic characteristics information 80
6.6.4.1 Profile character sets 81
6.6.4.2 Comments character sets 81
6.6.4.3 Alternative representation character sets 82
6.6.5 Fonts list 82
6.6.6 Document management attributes 82
6.6.6.1 Additional information 83

7 Specification of constituent constraints 84
7.1 Introduction 84
7.1.1 Diagrams of relationships of logical constituents 84
7.1.2 Notation 92
7.2 Document profile constituent constraints 93
7.2.1 Macro definitions 94
7.2.2 Constituent constraints 100
7.2.2.1 DocumentProfile 100
7.3 Logical constituent constraints 105
7.3.1 Macro definitions 105
7.3.2 Factor constraints 119
7.3.2.1 FACTOR ANY-LOGICAL 119
7.3.2.2 FACTOR COMP-LOGICAL 119
7.3.2.3 FACTOR BASIC-LOGICAL 119
7.3.2.4 FACTOR ANY-COMMON 120
7.3.3 Constituent constraints 120
7.3.3.1 DocumentLogicalRoot 120
7.3.3.2 Passage 120
7.3.3.3 NumberedSegment 121
7.3.3.4 Number 121
7.3.3.5 Paragraph 122
7.3.3.6 Phrase 122
7.3.3.7 Footnote 122
7.3.3.8 FootnoteNumber 123
7.3.3.9 FootnoteReference 123
7.3.3.10 FootnoteBody 124
7.3.3.11 FootnoteText 124
7.3.3.12 BodyText 124
7.3.3.13 Reference 125
7.3.3.14 ReferencedContent 125
7.3.3.15 BodyRaster 125
7.3.3.16 BodyGeometric 126
7.3.3.17 CommonContent 126
7.3.3.18 CommonText 126
7.3.3.19 CommonReference 127

7.3.3.20 CommonRaster . . . 127
7.3.3.21 CommonGeometric . . . 127
7.3.3.22 PageNumber . . . 128
7.3.3.23 TableNumber . . . 128
7.3.3.24 NumberedList . . . 128
7.3.3.25 UnNumberedList . . . 129
7.3.3.26 DefinitionList . . . 129
7.3.3.27 ListItem . . . 129
7.3.3.28 ListTerm . . . 130
7.3.3.29 Table . . . 130
7.3.3.30 Row . . . 130
7.3.3.31 EntryElement . . . 131
7.3.3.32 EntryText . . . 131
7.3.3.33 EntryRaster . . . 131
7.3.3.34 EntryGeometric . . . 132
7.4 Layout constituent constraints . . . 133
7.4.1 Macro definitions . . . 133
7.4.2 Factor constraints . . . 136
7.4.2.1 FACTOR ANY-LAYOUT . . . 136
7.4.2.2 FACTOR ANY-PAGE . . . 136
7.4.2.3 FACTOR ANY-FRAME-FIXED . . . 137
7.4.2.4 FACTOR ANY-FRAME-VARIABLE . . . 137
7.4.3 Constituent constraints . . . 138
7.4.3.1 DocumentLayoutRoot . . . 138
7.4.3.2 PageSet . . . 138
7.4.3.3 Page . . . 139
7.4.3.4 RectoPage . . . 139
7.4.3.5 VersoPage . . . 140
7.4.3.6 CompositeHeader . . . 140
7.4.3.7 CompositeFooter . . . 141
7.4.3.8 VariableCompositeBody . . . 141
7.4.3.9 ColumnFixed . . . 142
7.4.3.10 ColumnVariable . . . 143
7.4.3.11 SnakingColumns . . . 144
7.4.3.12 SynchronizedColumns . . . 145
7.4.3.13 BasicFloat . . . 146
7.4.3.14 FootnoteArea . . . 147
7.4.3.15 ArrangedContentFixed . . . 148
7.4.3.16 ArrangedContentVariable . . . 148
7.4.3.17 SourcedContentFixed . . . 149
7.4.3.18 SourcedContentVariable . . . 150
7.4.3.19 BasicHeader . . . 151
7.4.3.20 BasicFooter . . . 151
7.4.3.21 BasicBody . . . 152
7.4.3.22 GenericBlock . . . 152
7.4.3.23 SpecificBlock . . . 154
7.4.3.24 TableArea . . . 155
7.4.3.25 TableHeader . . . 156
7.4.3.26 TableLabel . . . 156

7.4.3.27 RowArea . . . 157
7.4.3.28 Cell . . . 158
7.4.3.29 TableLabelContent . . . 159
7.5 Layout style constituent constraints . . . 160
7.5.1 Macro definitions . . . 160
7.5.2 Factor constraints . . . 160
7.5.3 Constituent constraints . . . 160
7.5.3.1 LStyle1 . . . 160
7.5.3.2 LStyle2 . . . 160
7.5.3.3 LStyle3 . . . 161
7.5.3.4 LStyle4 . . . 161
7.5.3.5 LStyle5 . . . 162
7.5.3.6 LStyle6 . . . 162
7.5.3.7 LStyle7 . . . 163
7.5.3.8 LStyle8 . . . 163
7.5.3.9 LStyle9 . . . 164
7.5.3.10 LStyle10 . . . 164
7.5.3.11 LStyle11 . . . 164
7.5.3.12 LStyleT2 . . . 165
7.5.3.13 LStyleT4 . . . 165
7.5.3.14 LStyleT5 . . . 165
7.5.3.15 LStyleT8 . . . 166
7.5.3.16 LStyleT9 . . . 166
7.6 Presentation style constituent constraints . . . 166
7.6.1 Macro definitions . . . 166
7.6.2 Factor constraints . . . 166
7.6.3 Constituent constraints . . . 167
7.6.3.1 PStyle1 . . . 167
7.6.3.2 PStyle2 . . . 167
7.6.3.3 PStyle3 . . . 168
7.6.3.4 PStyle4 . . . 168
7.7 Content portion constituent constraints . . . 169
7.7.1 Macro definitions . . . 169
7.7.2 Factor constraints . . . 169
7.7.3 Constituent constraints . . . 169
7.7.3.1 Character-content-portion . . . 169
7.7.3.2 Raster-graphics-content-portion . . . 171
7.7.3.3 Geometric-graphics-content-portion . . . 171

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
8.2 Interchange format SDIF . . . 173
8.2.1 Interchange format . . . 173
8.2.2 Document application profile object identifier . . . 173
8.2.3 Encoding of application comments . . . 173

Annex A
Amendments and technical corrigenda 175
A.1 Amendments 175
A.1.1 Amendments to the base standard 175
A.1.2 Proposed changes to standards due to defects 176
A.2 Technical corrigenda 176
A.2.1 Technical corrigenda to this Specification 176
A.3 Versions of the ODA Specification 176

Annex B
Recommended practices 177
B.1 Transfer methods for ODA 177
B.1.1 Conveyance of ODA over CCITT X.400-1984 177
B.1.2 Conveyance of ODA over FTAM 177
B.1.3 Conveyance of ODA over DTAM 178
B.1.4 Conveyance of ODA over flexible disks 178
B.2 Font reference 178
B.3 ISO/IEC 8632 (CGM) constraints for this document application profile 179
B.3.1 Delimiter elements 180
B.3.2 Metafile descriptor elements 180
B.3.3 Picture descriptor elements 181
B.3.4 Control elements 181
B.3.5 Graphical primitive elements 181
B.3.6 Attribute elements 182
B.3.7 External elements 183
B.4 Interoperability with SGML applications 183

Annex C
Bibliography 184

List of figures

Figure 1 - Body layout types A, B and C with header and footer above and below the body area 31
Figure 2 - Header and footer frame layouts corresponding to Figure 1 31
Figure 3 - Body layout type C with header and footer to the right and left of the body area . 32
Figure 4 - Header and footer frame layouts corresponding to Figure 3 32
Figure 5 - Example of body area layout for body layout type A 34
Figure 6 - Example of body area layout for body layout type B 34
Figure 7 - Example of body area layout for body layout type C 34
Figure 8 - Example of the layout of a snaking columns frame 35
Figure 9 - Example of the layout of a synchronized column 37
Figure 10 - Example of a table 39
Figure 11 - Frames used to represent table labels 40
Figure 12 - Frames used to represent table rows 40
Figure 13 - Specification of margins 51
Figure 14 - Code extension features (basic case) 57
Figure 15 - Code extension features (all possible cases) 58
Figure 16 — DocumentLogicalRoot, 1st level 84
Figure 17 — DocumentLogicalRoot, 2nd level 85
Figure 18 — Paragraph and ListTerm 85
Figure 19 — Phrase 86
Figure 20 — Footnote 86
Figure 21 — Table 87
Figure 22 — Reference 87
Figure 23 — NumberedList 88
Figure 24 — UnNumberedList 88
Figure 25 — DefinitionList 89
Figure 26 — CommonContent 89
Figure 27 — DocumentLayoutRoot 89
Figure 28 — Page, RectoPage and VersoPage 90
Figure 29 — CompositeHeader and CompositeFooter 91
Figure 30 — VariableCompositeBody 91
Figure 31 — TableArea 92

List of tables

Table 1 — Nominal page sizes 29
Table 2 — Layout of tables 49
Table 3 — List of number string identifiers for l... 66
Table 4 — List of number string identifiers for... 67
Table 5 — Symbols to denote shift functions 93

Part 27 - Open Document Architecture Level 2 to Level 3 Migration DAP

Introduction

This Specification is defined within the context of Functional Standardization, in accordance with the principles specified by ISO/IEC TR 10000, "Framework and Taxonomy of International Standardized Profiles". The context of Functional Standardization is one part of the overall field of Inforamtion Technology (IT) standardization activities, covering base standards, profiles and registration mechanisms. AProfile defines a combination of base standards that collectively perform a specific well-defined IT Function. Profiles standardize the use of options and other variations in the base standards, and provide a basis for development of uniform, internationally recognized conformance test suites.

One of the most important roles for an ISP is to serve as the basis for the development (by organizations other than ISO and IEC) of internationally recognized tests and test centers. ISPs are produced not simply to "legitimize" a particular choice of base standards and options, but to promote real systems interoperability. The development and widespread acceptance of tests based on this and other ISPs is crucial to the successful realization of this goal.

Development of this document application profile has been done in close cooperation between several organizations. These include ODA expert groups within the Asia-Oceania Workshop (AOW), CCITT Study Group VIII, European Workshop for Open Systems (EWOS) and 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 ISP is harmonized among the three workshops and has been ratified by the three workshops' plenary assemblies.

The purpose of this Specification is to facilitate the interworking of applications interchanging documents based on CCITT Rec. T.410 series | ISO 8613, ODA. This Specification 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 Specification is in accordance with the standardized proforma and notation defined in Amendment 1 - Annex F of CCITT Rec. T.411 | ISO 8613-1.

1 Scope

This Specification 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 Specification range from simple documents to highly structured technical reports, articles and typeset documents such as brochures. This Specification provides a comprehensive level of features for the transfer of documents between these systems.

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

— formatted form;

— processable form;

— formatted processable form.

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

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

A document structured in accordance with this Specification is represented for interchange by one of two document application profiles. One document application profile uses the Office Document Interchange Format (ODIF), the other document application profile uses the Office Document Language (ODL), both as defined in CCITT Rec. T.415 | ISO 8613-5.

When this document refers to *this Specification,* it is refering to either of the document application profiles defined by this Specification.

2 Normative references

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

2.1 Paired Recommendations | International Standards equivalent in technical content

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

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

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

ISO 8613-1 : 1989/Add.[1]: *Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 1: Introduction and general principles - Addendum 1 : Document Application Profile Proforma and Notation*.

— 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*

[1] To be published.

Architecture (ODA) and interchange format - Part 5: Office Document interchange format (ODIF).

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

ISO 8613-6 : 1989, *Information processing - Text and office systems - Office Document Architecture (ODA) and interchange format - Part 6: Character content 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 - 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 (1992), *Document Application Profile PM-11 for the Interchange of Character Content Documents in Processable and Formatted Forms*.

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

— CCITT Recommendation T.506[2], *Document Application Profile PM-36/FOD36 - Open Document Format : Extended document structure - Character, raster graphics and geometric graphics content architectures*.

ISO/IEC ISP 11182-1 : 1993, *Information technology - International Standardized Profile*

[2] To be published.

FOD36 - Open Document Format : Extended document structure - Character, raster graphics and geometric graphics content architectures - Part 1 : Document Application Profile (DAP).

NOTE - In the titles of the various parts of ISO 8613 the term *Office* occurs. ISO has instructed that this be changed to *Open*, and the new edition of ISO 8613 (to be published) will use *Open* in place of *Office*. For this reason this Specification uses the term *Open* in its title.

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 : 1991, *Information technology - Registration of repertoires of graphic characters from ISO/IEC 10367*.

— ISO/IEC 8632 : 1992, *Information technology - Computer graphics - Metafile for the storage and transfer of picture description information*
  - *Part 1: Functional specification*
  - *Part 3: Binary encoding*.

— ISO 8859-1 : 1987, *Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1*.

— ISO 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 interchange*.

— ISO/IEC TR 10000-1 : 1992, *Information technology - Framework and taxonomy of International Standardized Profiles - Part 1: Framework*.

— ISO/IEC TR 10000-2 : 1992, *Information technology - Framework and taxonomy of International Standardized Profiles - Part 2: Taxonomy of OSI profiles*.[3]

3 Definitions

For the purposes of this Specification, the following definitions apply.

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

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

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 Specification are BodyText, Footnote and RectoPage.

In clause 6, each constituent constraint provided by this Specification 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 Specification.

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 Specification, the constituents may be known to the user by different names.

[3] To be published.

4 Relationship with other profiles

This Specification belongs to a series of hierarchically related profiles which include FOD11 | PM-11, FOD26 | PM-26 and FOD36 | PM-36.

The features supported by this Specification are a superset of the features supported by the profiles FOD11 | PM-11 and FOD26 | PM-26 and thus all data streams that are conformant to FOD11 | PM-11 and FOD26 | PM-26 are also conformant to this Specification, apart from the DAP identifier.

Also, the features supported by this Specification are a subset of the features supported by the profile FOD36 | PM-36 and thus all data streams that are conformant to this Specification are also conformant to FOD36 | PM-36, apart from the DAP identifier.

5 Conformance

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

5.1 Data stream conformance

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

a) the data stream shall be encoded either in accordance with the ASN.1 encoding rules defined in CCITT Rec. X.209 | ISO 8825 or the SGML encoding rules defined in ISO 8879;

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 Specification; 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 Specification;

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 Specification.

A conforming receiving implementation shall be capable of receiving either any data streams conforming to this Specification structured in accordance with Office Document Interchange Format (ODIF), or any data streams conforming to this Specification structured in accordance with Office Document Language (ODL) or both of these. Receiving usually, but not always, involves recognizing and further processing the data stream elements. The explicit meaning of "receiving" is determined by another part of this Specification.

A receiving system which claims conformance to this document application profile shall be capable of handling data streams which are conformant to document application profiles that are subordinate to this document application profile within the taxonomy described in clause 4, i.e. FOD11 | PM-11 and FOD26 | PM-26.

FOD36 | PM-36 documents which adhere to this Specification can be distinguished from fully-functional FOD36 | PM-36 documents using the document profile document management attribute "Additional-information". Documents which adhere to this Specification are required to specify the value "Level 2 to Level 3 Migration DAP" for this attribute.

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 Specification. This clause also describes how these characteristics are represented in terms of constituent constraints.

6.1 Overview

6.1.1 General

This Specification 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 Specification 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 Specification. 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 Specification include all those referenced and contained in resource-documents and external-documents, if any (see 6.6.1 and 6.6.2).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.

6.1.2 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.

6.1.3 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.

6.1.4 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.

6.1.5 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 Specification to represent the characteristics of documents containing logical component descriptions.

Different constituent constraints may be used to represent and distinguish parts of a document that have

different logical characteristics. This subclause describes the general characteristics and typical uses of the constituent constraints that are provided.

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

6.2.2 Overview of the logical structure

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

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

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

6.2.3 Body part of the logical structure

6.2.3.1 DocumentLogicalRoot

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

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

6.2.3.2 Passage

Passage is a constituent constraint that represents the first level of logical subdivision of a document. 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;

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

The immediate subordinates of a Passage consist of an optional arbitrary ordered sequence of one or more

of the following constituent constraints:

— Paragraph;

— BodyGeometric;

— BodyRaster;

— BodyText;

— Table;

— NumberedList;

— UnNumberedList;

— DefinitionList;

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

A Passage shall have at least one of the above constituent constraints as a subordinate.
A document may contain several different class definitions of the type Passage, each of which defines the common characteristics of sets of Passages within the document, such as their allowed subordinates or layout properties. For example, a class of Passages may be defined which always 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 Passage or another higher level 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, Table, NumberedList and Footnote may be initialized on a Passage.

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

— Paragraph;

— BodyGeometric;

— BodyRaster;

— BodyText;

— Table;

— NumberedList;

— UnNumberedList;

— DefinitionList;

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

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

6.2.3.4 Number

Number is a constituent constraint that represents the identifier of a NumberedSegment 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;
Reference;
— Footnote.

—

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.

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;

— ListTerm;

— UnNumberedList.

In addition, BodyText may be specified as a subordinate to Phrase and Reference. 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 the layout category name 'Footnote'. 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;

— ListTerm;

— Phrase.

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 the attribute layout category with the value 'Footnote' for this constituent constraint. This along with a "permitted categories" attribute with the value 'Footnote' 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 the attribute layout category with the value 'Footnote' for this constituent constraint. This along with a "permitted categories" attribute with the value 'Footnote' 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 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. In this profile, a Phrase may be used only as a subdivision of constituents constraints of type ListItem, so that a ListItem may contain embedded footnotes and references to other parts of the document content.

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

— BodyText;

— 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.

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.

6.2.3.12.1 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;

— ListTerm;

— Phrase;

— 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 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.

6.2.3.12.2 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 6.2.3.13.1).

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 the attribute layout category with the value 'Footnote' for this constituent constraint. This along with a "permitted categories" attribute with the value 'Footnote' 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.

6.2.3.13 Constituents used to represent tables

6.2.3.13.1 Introduction

For the purpose of this Specification, 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 consists of a sequence of one or more cells.

The subclauses below define the logical constituents used to represent tables. Figure 21 illustrates the structural relationships between the constituents used to represent a table. Subclause 6.4.1.3.7 describes how tables are intended to be laid out.

6.2.3.13.2 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.

6.2.3.13.3 Row

Row is a constituent constraint that is a subordinate of the constituent constraint Table and represents a row of entries in a table.

The immediate subordinates of a Row consist of a sequence of constituent constraints EntryElement.

6.2.3.13.4 EntryElement

EntryElement is a constituent constraint that represents a single entry in a table. It is a sub-division of constituent constraints of type Table. It is specified as a subordinate to a Row and represents a single entry in a table.

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.

6.2.3.13.5 EntryText, EntryRaster and EntryGeometric

EntryText, *EntryRaster* and *EntryGeometric* are constituent constraints which represent content that is to be entered into tables. 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 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.

6.2.3.14 Constituents representing lists

6.2.3.14.1 Introduction

This Specification 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.

6.2.3.14.2 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.

6.2.3.14.3 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.

6.2.3.14.4 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.

6.2.3.14.5 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.

6.2.3.14.6 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;

— Reference;

— Footnote.

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

— TableNumber;

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 the header or footer area of a page. For example, header or footer content that appears on each page in a sequence of pages may be represented by this constituent constraint.

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

6.2.4.3 PageNumber

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

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

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

TableNumber is a constituent constraint that represents the common character content that is to be laid out in the header or footer area of 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.

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

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

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

6.2.1 Overview of the layout characteristics

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

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

Three body layout types are provided by this Specification. 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 Specification 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 sub-divided into areas to support different layout requirements. These features are described further in 6.3.5 and 6.3.6.

6.2.2 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.

6.2.3 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.

6.2.4 Page characteristics

6.2.4.1 Page constituents

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

The only difference in the characteristics of these types of constituent constraints concerns the values that may be specified for the parameter "side of sheet" in the attribute "medium type". In the case of Page, 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 pages that make up a page set consist of an optional initial page which is represented by a constituent constraint Page and which is optionally followed by either:

— A sequence of pages represented by the constituent constraint Page. All pages in this sequence shall have the same layout characteristics but these characteristics may differ from those on the initial page.

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

A page set shall contain at least one page.

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

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

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).

6.2.4.2 Page dimensions

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

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

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

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

6.2.4.3 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 specified 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.

6.2.4.4 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

Table 1 — Nominal page sizes

Page type	Size in inches or millimeters	Size in BMUs	ARA in BMUs
ISO A5	148mm x 210mm	7 015 x 9 920	not defined
ISO A4	210mm x 297mm	9 920 x 14 030	9 240 x 13 200
ISO A3	297mm x 420mm	14 030 x 19 840	13 200 x 18 480
ANSI legal	8,5in. x 14in.	10 200 x 16 800	9 240 x 15 480
ANSI A	8,5in. x 11in.	10 200 x 13 200	9 240 x 12 400
ANSI B	11in. x 17in.	13 200 x 20 400	12 744 x 19 656
Japan-legal	257mm x 364mm	12 141 x 17 196	11 200 x 15 300
Japan-letter	182mm x 257mm	8 598 x 12 141	7 600 x 10 200

default page offset may be specified in the document profile.

6.2.4.5 Page layout characteristics

6.2.4.5.1 General characteristics

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

— a body area, which is reserved for content that belongs to the body part of the document (see 6.3.5);

— a header area, which is reserved for common header content (see 6.3.6);

— a footer area, which is reserved for common footer content (see 6.3.6).

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

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

6.2.4.5.2 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.

6.2.4.5.3 Header/footer area layout

Two 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. This header/footer layout applies to body layout types A, B and C;

— 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. This header/footer layout applies to body layout type C only.

6.2.4.5.4 Page layout

Page layout type is determined by a combination of the body layout type and the H/F layout type. The following combinations are permitted:

— Body layout type A and H/F layout A1 (equivalent to Page Layout A of FOD26);

— Body layout type B and H/F layout A1 (equivalent to Page Layout B of FOD26);

— Body layout type C and H/F layout A1 (equivalent to Page layout C of FOD26);

— Body layout type C and H/F layout B1 (equivalent to Page layout D of FOD26);

It is intended that all applications which use this Specification shall support the combinations of body layout A and H/F layout A1, whereas support for other combinations may be specified as optional.

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

The header and footer of H/F layout B1 is laid out to the right and left of the body area. Figure 3 illustrates

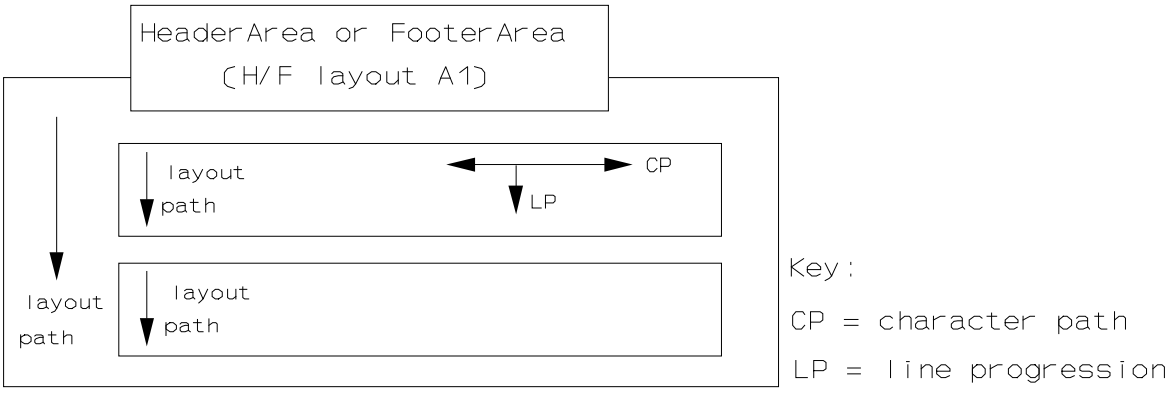

Figure 2 - Header and footer frame layouts corresponding to Figure 1

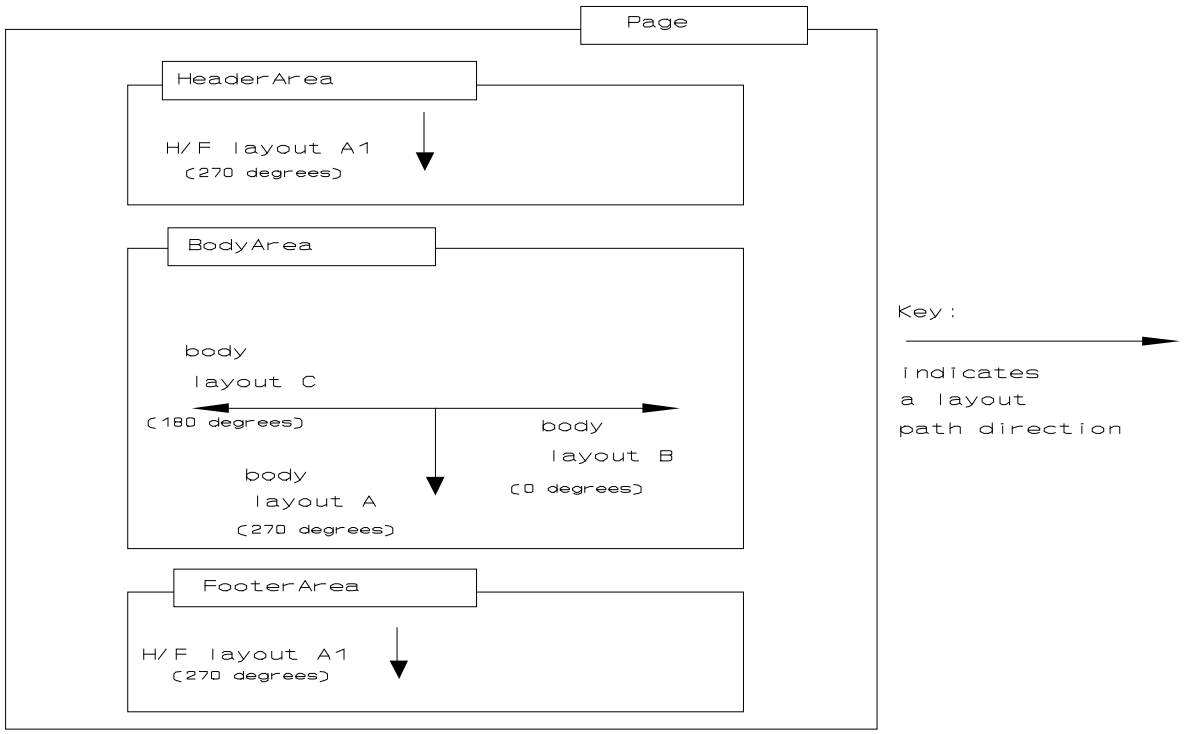

Figure 1 - Body layout types A, B and C with header and footer above and below the body area

this case, and Figure 4 illustrates H/F layout type B1 corresponding to this case.

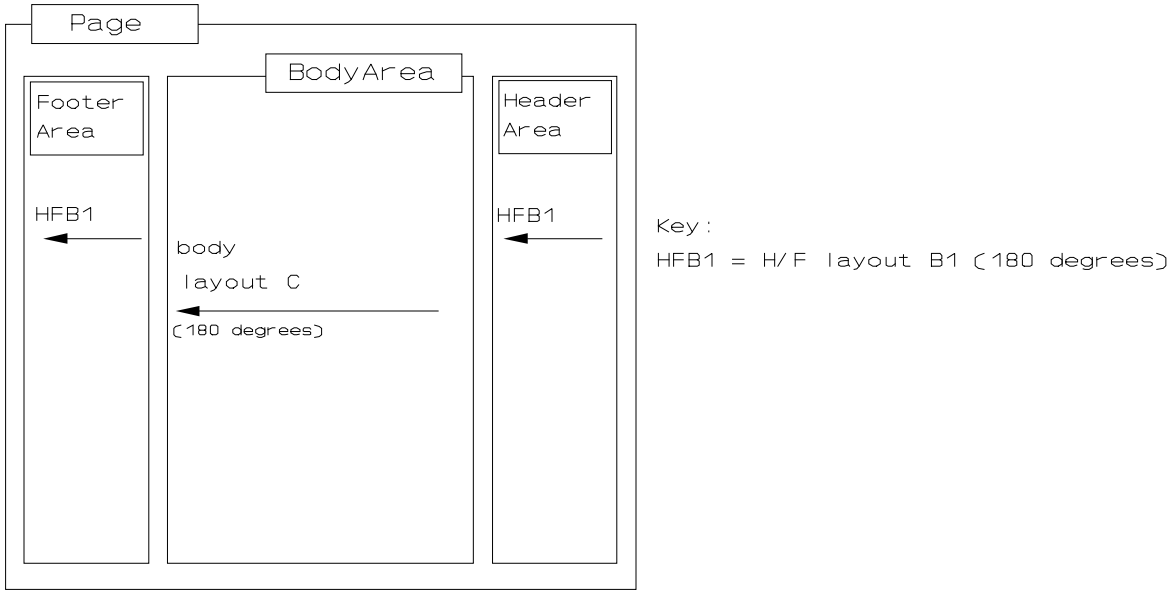

Figure 3 - Body layout type C with header and footer to the right and left of the body area

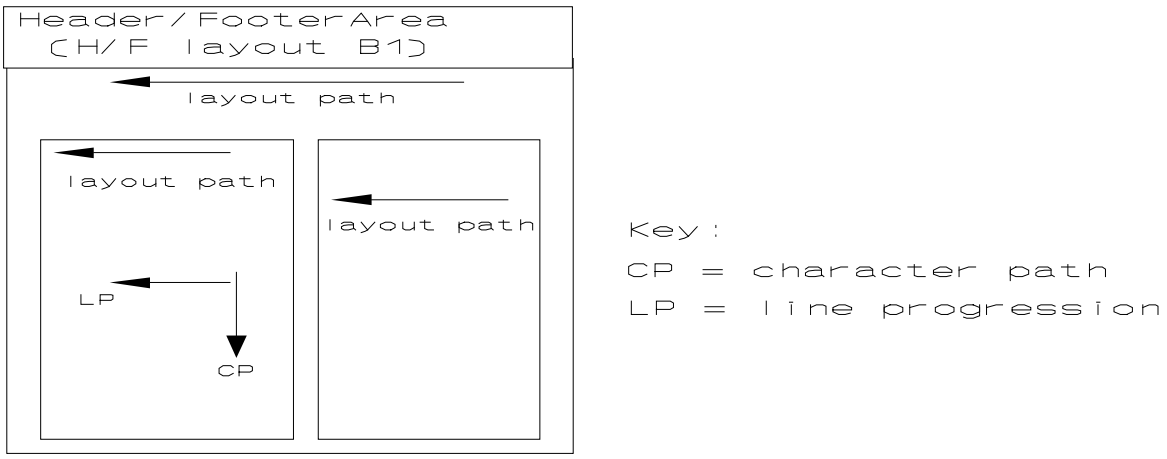

Figure 4 - Header and footer frame layouts corresponding to Figure 3

6.2.5 Body area characteristics

6.2.5.1 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 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.

6.2.5.2 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).

6.2.5.3 VariableCompositeBody

VariableCompositeBody is a constituent constraint that defines a composite frame which represents the entire body area, and which contains one or more subordinate variably positioned frames. These subordinates are not pre-determined and are automatically adjusted during the layout process to accommodate the content that is allocated to them, for example, to provide for combinations of single or multicolumn layout and the layout of footnotes.

A VariableCompositeBody frame has a fixed position and fixed dimensions. The layout path specified for this frame depends upon the 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;

— TableArea.

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

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

The layout path for VariableCompositeBody frames may be specified as 270, 0 or 180 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.

Except for SnakingColumns, all frames are constrained to have the same layout path as the VariableCompositeBody frame to which they are subordinate (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.

6.2.5.4 BasicFloat

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

This is a variably positioned frame which may be specified as a subordinate to frames of type VariableCompositeBody.

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

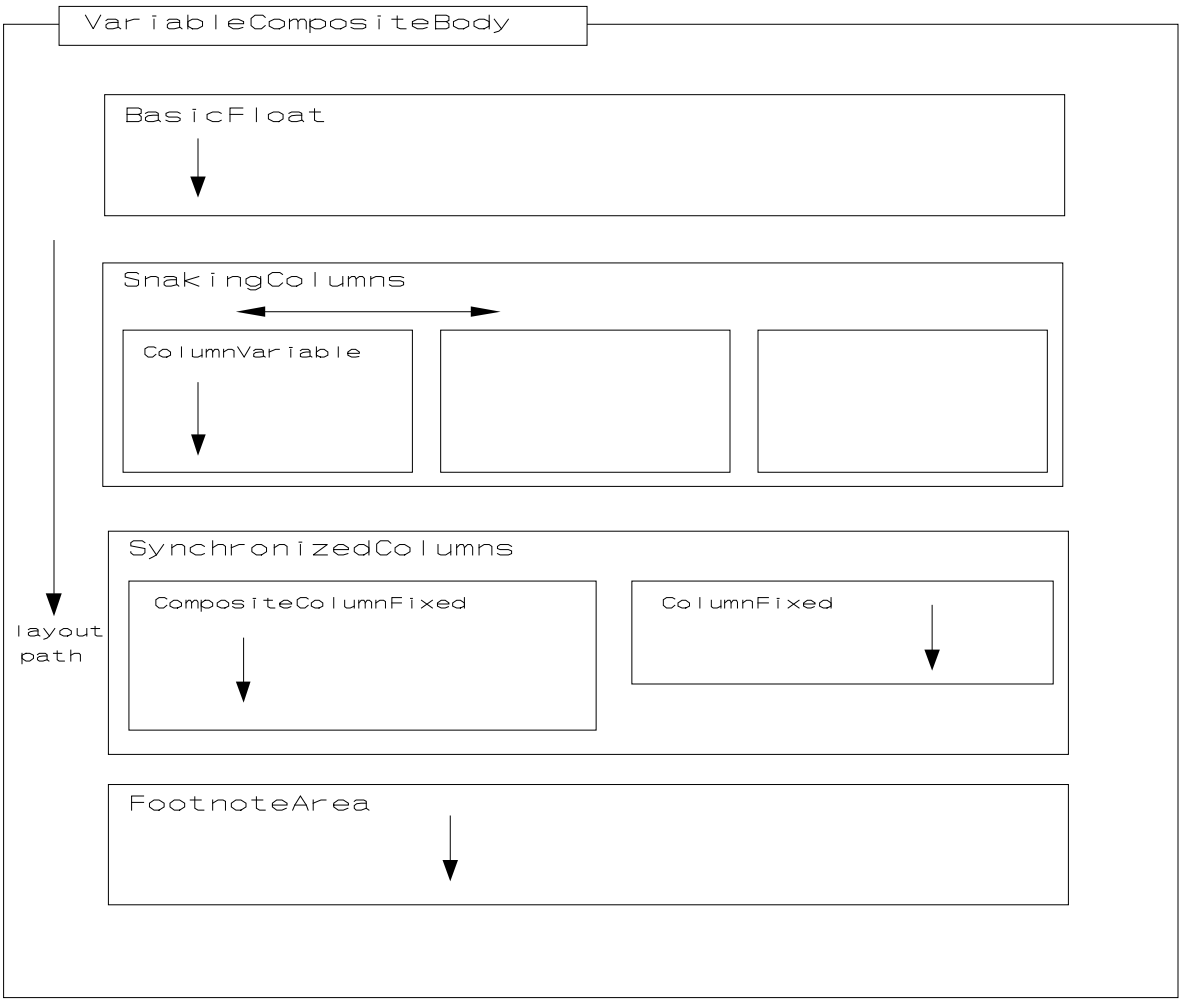

Figure 5 - Example of body area layout for body layout type A

be the minimum required to contain all the content allocated to the frame.

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

6.2.5.5 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.

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

Its immediate subordinates consist of one or more frames of the type ColumnVariable.

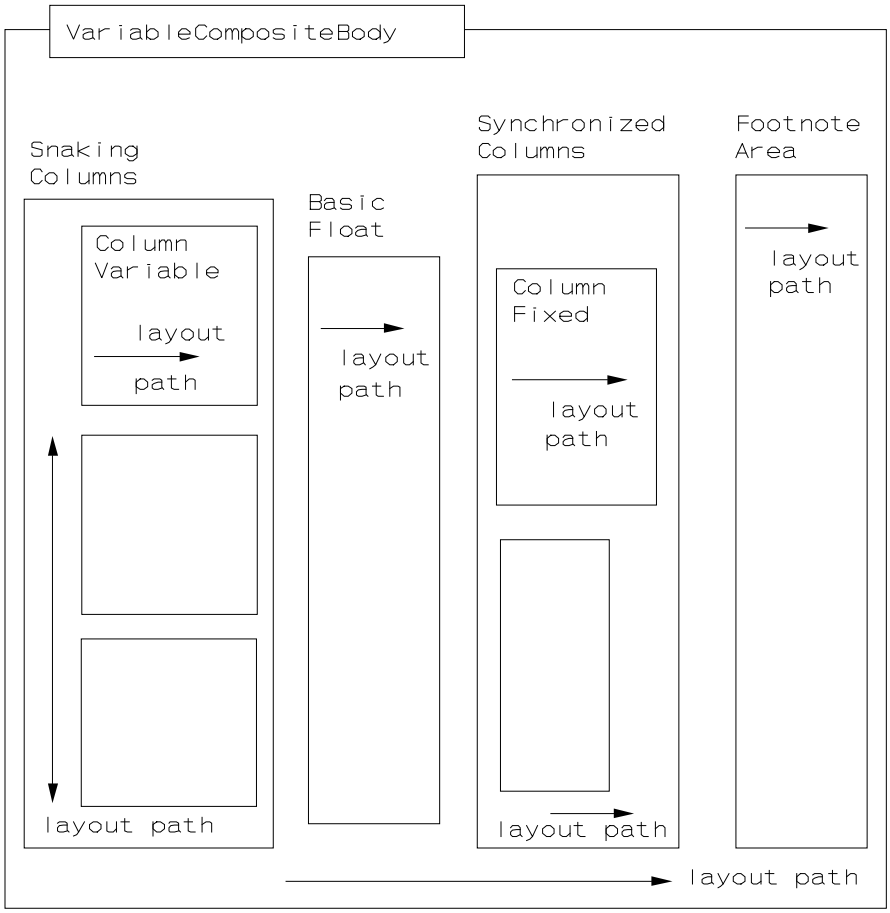

Figure 6 - Example of body area layout for body layout type B

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.

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.

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

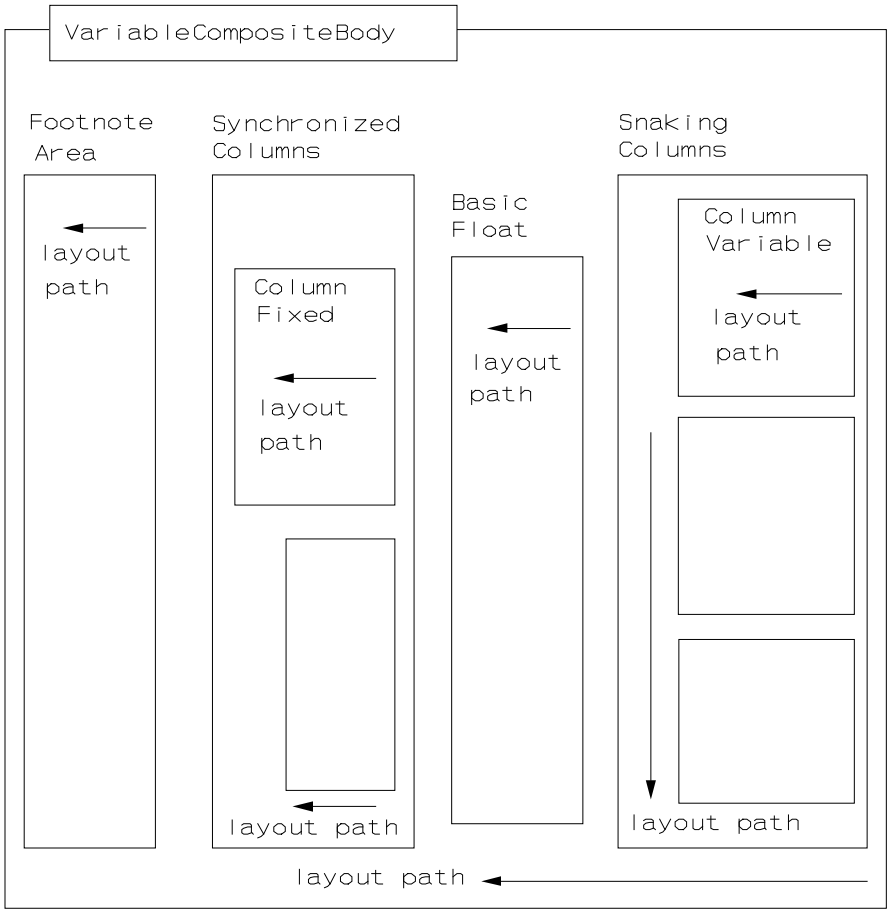

Figure 7 - Example of body area layout for body layout type C

6.2.5.6 SynchronizedColumns

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

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

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

The immediate subordinates of a SynchronizedColumns frame consist of an arbitrary ordered sequence of any number of frames of the type ColumnFixed.

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.

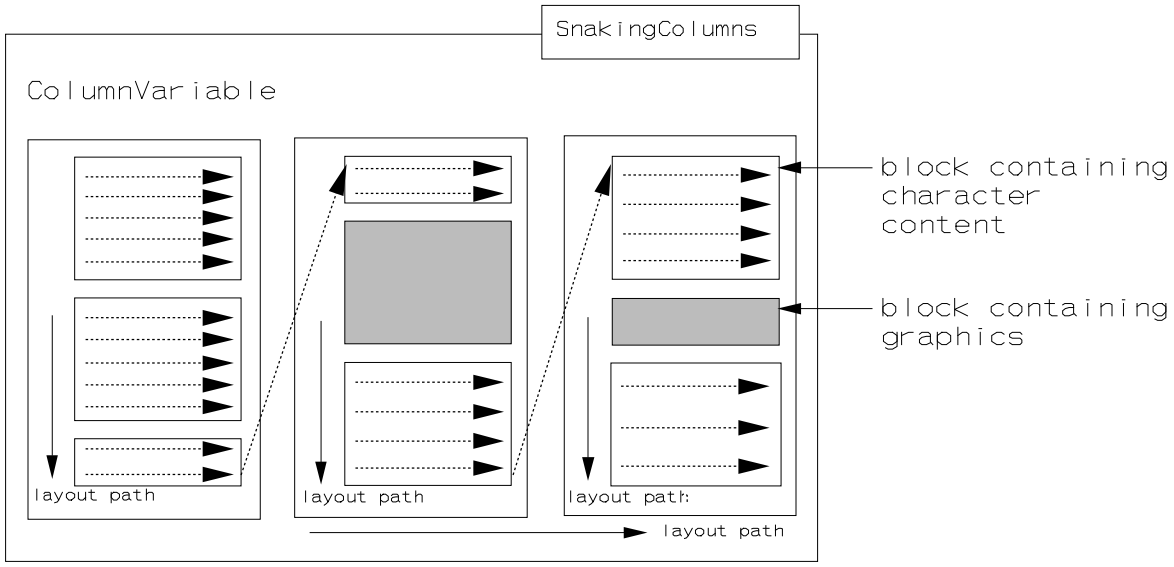

Figure 8 - Example of the layout of a snaking columns frame

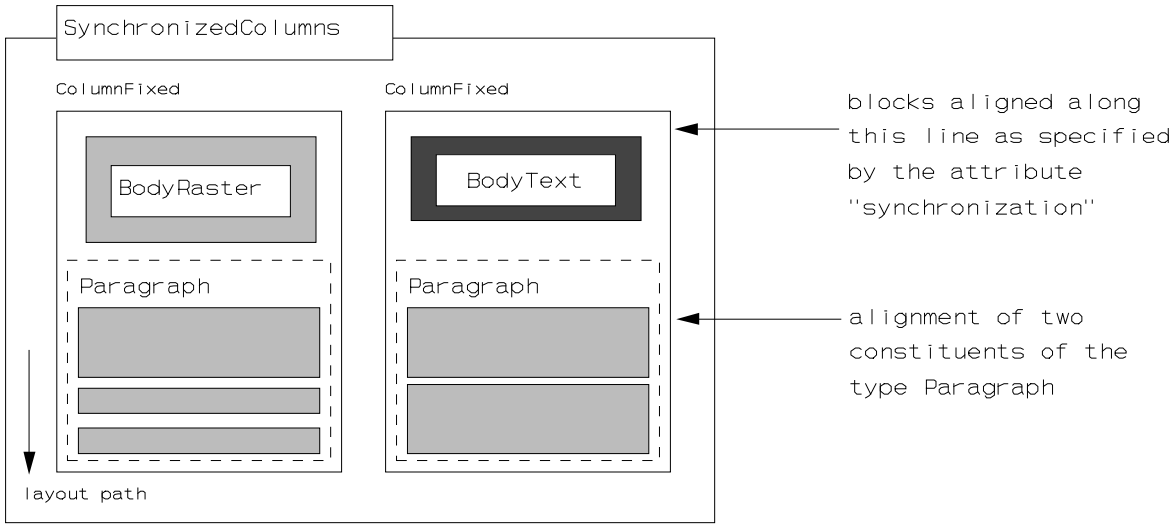

Figure 9 - Example of the layout of a synchronized column

6.2.5.7 ColumnVariable

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

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

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

The layout path for ColumnVariable frames is 270 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.

6.2.5.8 ColumnFixed

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

The dimension of this frame in the direction orthogonal to the layout path of the superior frame (that is, 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 (that is, 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.

6.2.5.9 FootnoteArea

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

This frame may be specified as a subordinate to VariableCompositeBody frames.

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 the VariableCompositeBody frame.

The dimension of FootnoteArea frames in the direction orthogonal to the layout path of its superior frame is specified by one of the sub-parameters "fixed-dimension" or "maximum size". In the direction of the layout path, the dimension is specified by 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 body area.

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

6.2.5.10 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 6.3.5.10.1 and the subsequent clauses define the individual constituents provided.

6.2.5.10.1 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.

An example of a table is shown in Figure 10. 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.

The frames that are used to represent table label and row areas are shown in Figures 11 and 12 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.

The frames which specifies the complete table area (that is, 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 6.4.1.3.7.

6.2.5.10.2 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 VariableCompositeBody constituent constraint.

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

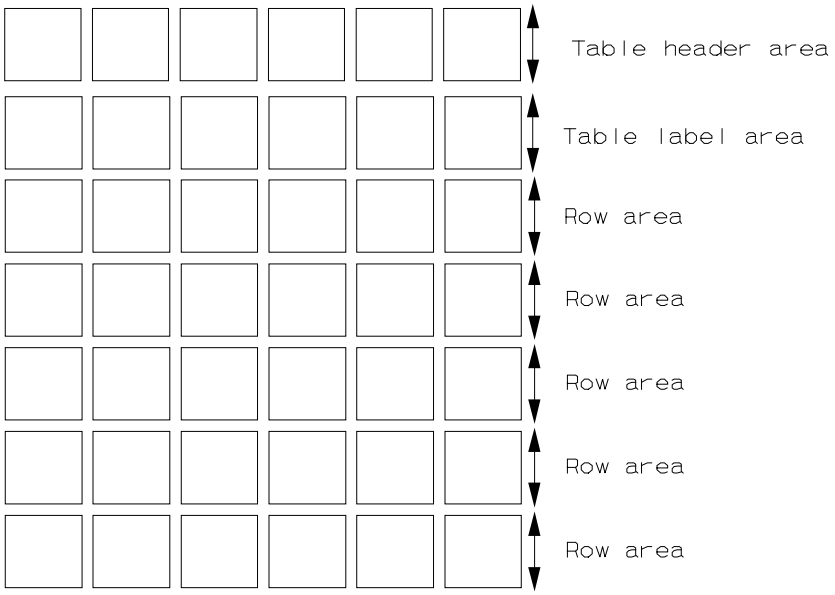

Figure 10 - Example of a table

the sub-parameter "Rule-B". Its layout path is specified as 270 degrees.

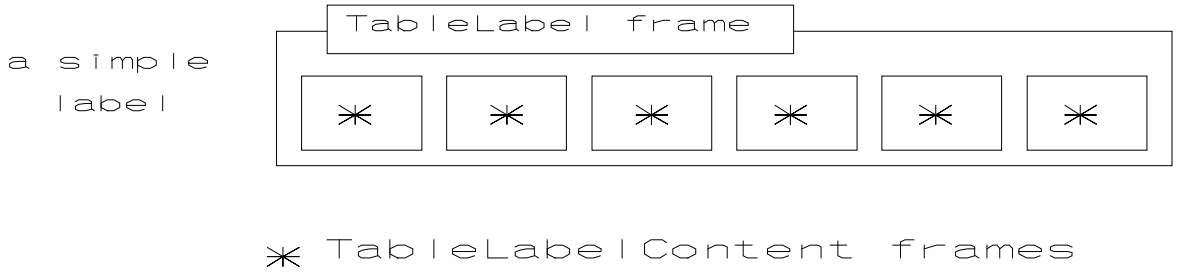

Figure 11 - Frames used to represent table labels

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.

6.2.5.10.3 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.

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

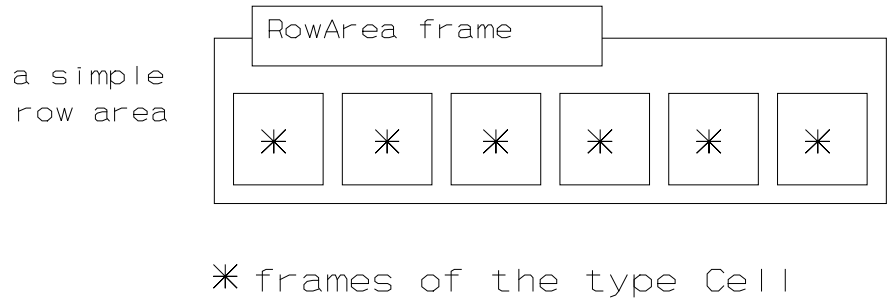

Figure 12 - Frames used to represent table rows

logical constituent constraints of the type CommonContent.

6.2.5.10.4 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 a sequence of one or more constituent constraints of the type TableLabelContent.

6.2.5.10.5 TableLabelContent

TableLabelContent is a constituent constraint that defines a lowest level frame that defines an area with in a TableLabel 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.

6.2.5.10.6 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 a sequence of one or more constituent

constraints of the type Cell.

6.2.5.10.7 Cell

Cell is a constituent constraint that defines a lowest level frame that specifies an area within a RowArea 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.

6.2.6 Header and footer area characteristics

6.2.6.1 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, t he 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.

6.2.6.2 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 pa ge and the layout paths that may be specified for them depends 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.

6.2.6.3 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 pa ge and the layout paths that may be specified for them depends 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.

6.2.6.4 SourcedContentVariable

A *SourcedContentVariable* frame is a constituent constraint that defines a lowest level frame that repres ents a region within a header or footer area that contains common content derived from the generic logi cal 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 header or footer area. It is the immediate subordinate of the frame of the constituent constraint type CompositeHeader or CompositeFooter.

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.

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

The layout path of this frame is the same as that of the containing header or footer area:

- the layout path of the frame is:
  * 270 degrees for H/F layout A1;
  * 180 degrees for H/F layout B1 (see 6.3.4.5 and the comment in 7.4.3.18)
- the horizontal dimension of the frame is:
  * either fixed or specified by sub-parameter "maximum-size" for H/F layout A1;
  * 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 layout A1;
  * either fixed or specified by sub-parameter "maximum-size" for H/F layout B1.

Typically, this frame is used for the positioning of content which is generated during the layout process,

such as a character sequence containing a page number.

6.2.6.5 ArrangedContentVariable

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

This frame 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.

6.2.6.6 SourcedContentFixed

A *SourcedContentFixed* frame is a constituent constraint that defines a lowest level frame that represen ts a region within a header or footer area that contains common content derived from the generic logical structure. This frame has a fixed position and dimensions.

This frame is required to specify the attribute "logical source" which indicates the particular instance of t he 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 fram e of the constituent constraint type TableHeader.

The layout path of this frame is equal to that of the containing header or footer area or TableHeader ar ea.

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.

6.2.6.7 ArrangedContentFixed

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

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

6.2.7 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 this type are restricted to occur within the header and footer areas of the page.

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

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 gene ric layout structure is not present, then these mechanisms are restricted as described in 6.4.3.

6.3.1 Flow controls

Various mechanisms are provided to control the allocation of constituent constraints representing the bo dy 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.

6.3.1.1 Allocation of content to page sets

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

— layout in a nominated page set;

— starting a new page set.

The first method provides the ability to specify that a part of a document is to be laid out entirely within a specified page set. This may be specified for constituent constraints of the types Passage, NumberedSegment, Paragraph, 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 logic al 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;

— 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.

6.3.1.2 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;

— 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 identifi er of the required page class.

6.3.1.3 Allocation of content to body areas

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

If the page contains a composite body area, i.e., a VariableCompositeBody 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 provid ed for the layout of logical constituent constraints of the type Table (see 6.4.1.3.7). Also, the layout of

footnotes is described in 6.4.1.3.8.

6.3.1.3.1 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 th at at the beginning of the succeeding page. When continuation to the same type of area in a succeedi ng page is not possible because of conflict with other layout directives or because the attribute "generat or 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. In the case of single or synchronized column areas, the attribute "new layout object" may indicate the category name corresponding to the frame class of the single column ar ea or of any of the columns within the synchronized column area that is required.

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

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.

6.3.1.3.2 Layout into BasicFloat 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.

6.3.1.3.3 Layout in SnakingColumns frames

A SnakingColumns frame is a composite frame which contains columns represented by lowest level fra mes.
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 t he superior frame and the layout process will continue automatically in the next column.

6.3.1.3.4 Layout in SynchronizedColumns frames

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

6.3.1.3.5 Layout of tables

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

Table 2 illustrates the relationships between the logical constituent constraint Table and its subordinates, and the frames used to lay out these constituent constraints.

Table 2 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 RowAre a.

The layout mechanism is achieved for the logical constituent constraints Table and Row by specifying t he attribute "new layout object" with a value indicating the identifier of the required frame class of the ty pe TableArea and RowArea 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 of the affected EntryElement is constructed using SEQuence, the attribute "new layout object" is used t o 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 of the affected EntryElement is constructed by REPetition, the value of the attribute "new layout object"

Table 2 — Layout of tables

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

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.

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 o ver two successive frames or pages.

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 are derived from logical constituent constraints of the type CommonContent in the generic logical structure using the logical source mechan ism.

6.3.1.3.6 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 o ut 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 t he types FootnoteNumber and FootnoteText, and layout constituent constraints of the type FootnoteArea are all required to have the category name 'Footnote'.

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

If the content belonging to a footnote cannot all be accommodated in the footnote area on one page, th en 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".

6.3.1.4 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 t he constituent constraint of the type CommonContent that is to be laid out in that area. The basic logic al 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.

6.3.2 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.

6.3.2.1 Margins

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

Margins may be specified for any constituent constraint representing a basic logical object; different mar gin 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 conte nt is to be laid out (see Figure 13).

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

6.3.2.2 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 a nd the previous one, if any, when they are laid out. Both may be specified for basic logical components

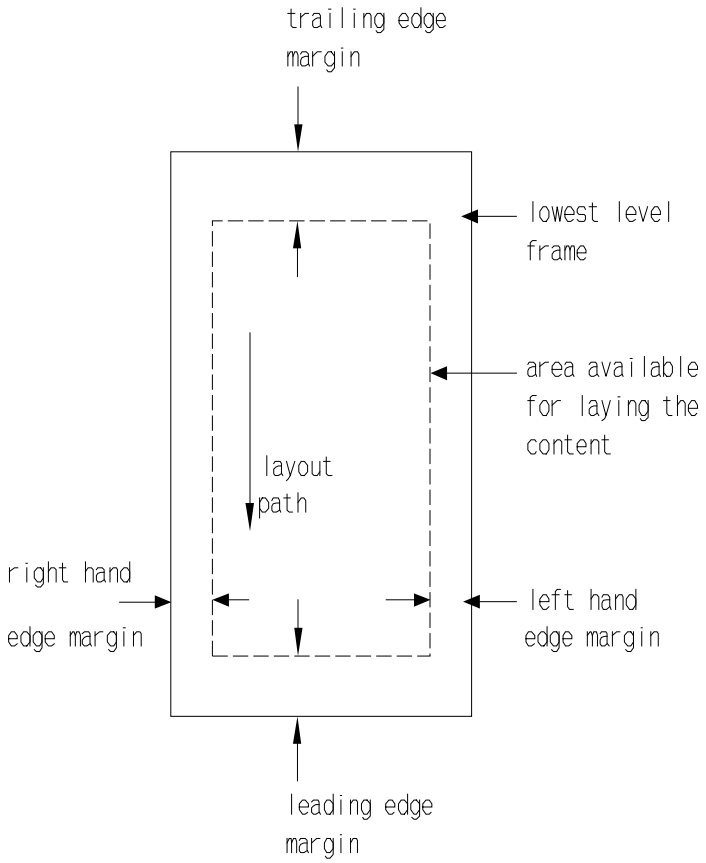

Figure 13 - Specification of margins

of any constituent constraint types. These distances are specified in BMUs by the attribute "separation". If no value is specified, then the minimum distance is assumed to be 0 BMUs.

6.3.2.3 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 with in a page. It may be specified for constituent constraints of the types Passage, NumberedSegment, Number, Paragraph, Phrase, Reference, ListItem, ListTerm, FootnoteText, ReferencedContent, FootnoteBody, Artwork, EntryElement, Row, Table, UnNumberedList, NumberedList, DefinitionList, FootnoteReference and BodyText. The attribute "indivisibility" is used to specify this feature.

6.3.2.4 Same layout object

Same layout object provides the means to specify that the start of the content associated with a logical object and the end of the content associated with the previous logical object are to be laid out within a single layout object. This may be specified for logical objects of the types NumberedSegment, Number, Paragraph, Phrase, Footnote, FootnoteBody, FootnoteReference, ReferencedContent, Reference, Table, NumberedList, UnNumberedList, DefinitionList, ListItem, ListTerm, BodyText, BodyRaster and

BodyGeometric. The attribute "same layout object" is used to specify this feature.

6.3.2.5 Concatenation

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

6.3.2.6 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 alig ned 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, PageNumber, TableNumber, FootnoteNumber, FootnoteText, FootnoteReference, CommonReference, ReferencedContent and CommonText, and when they contain formatted character content, BodyRaster, and BodyGeometric, EntryRaster, EntryGeometric, CommonRaster and CommonGeometric.

6.3.3 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 a re 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 t o 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 indic ate 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 i n the header/footer areas of each page may be distinguished from each other by mean s 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 i n 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 whethe

r 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 attribu te "application comments". Footnotes are intended to be read and laid out separately fr om 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 (se e 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 layou t areas, such as single or multiple column areas.

6.4 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 ras ter graphics and formatted processable geometric graphics. Any of these may be specified as the defa ult in the document profile.

6.4.1 Character content

6.4.1.1 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 imag e, positive image, primary font, 1st alternative font, 2nd alternative font, 3rd alternative f ont, 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;

— proportional line spacing;

— formatting indicator.

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

6.4.1.2 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 compon ent.

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 subseque nt editing.

6.4.1.3 Character repertoires

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

Any other graphic character set which is registered in accordance with ISO 2375 may be designated and invoked at any point in the document provided its use is indicated in the document profile as a non-basic value using the character presentation feature "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 h and 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/IEC 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 Specification is not the standard default value specifie d 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 Recommendations T.50 and T.51 and new CCITT Recommendation T.52 are under preparation. CCITT Recommendations T.50 and T.51 are intended to be completely compatible with ISO/IEC 646 (ISO-IR 6) and ISO 6937 (under revision) respectively.

6.4.1.4 Code extension techniques

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

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

b) G1, G2, G3 sets: no restrictions are placed on the character sets that may be designate d 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 grap hic character set is invoked into GR;

The code extension techniques allowed are illustrated in Figures 14 and 15.

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 8613-6,416 | 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 "gra phic 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.

6.4.1.5 Line spacing

Any value of line spacing may be specified. Values of 100, 150, 200, 300, 400 and 600 BMUs are basi c; 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 u sing 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).

6.4.1.6 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 u sing 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 Specification for use with Chinese characters.

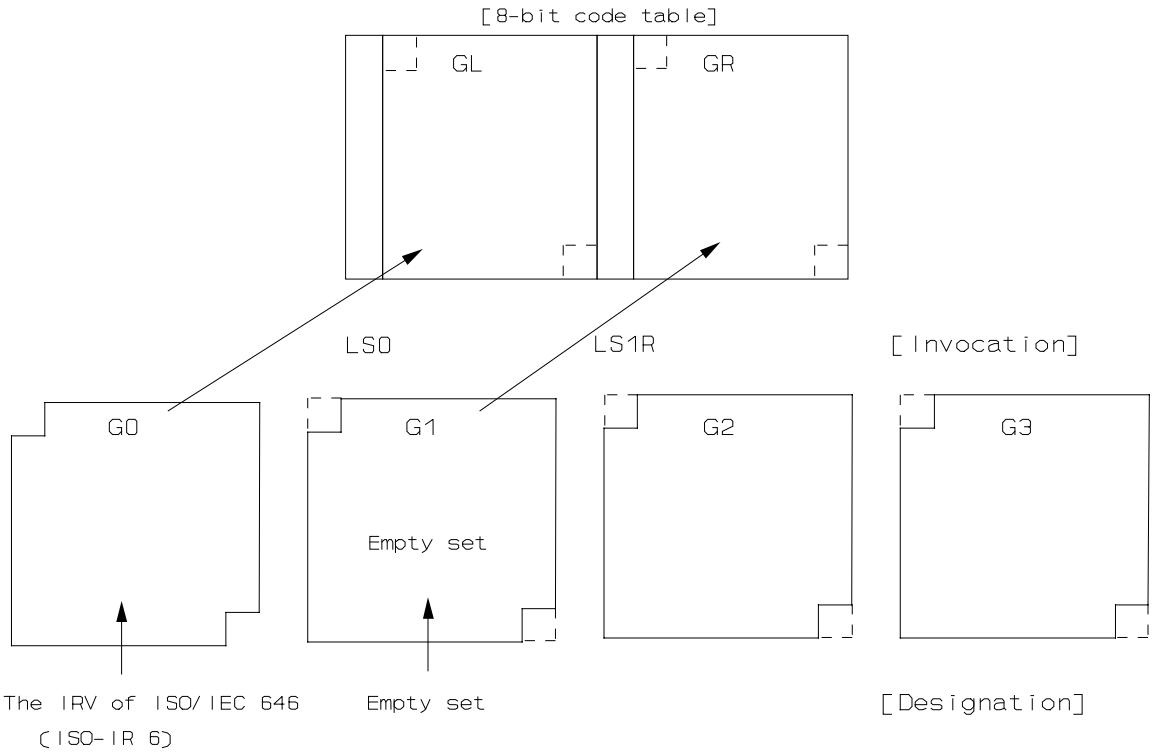

Figure 14 - Code extension features (basic case)

6.4.1.7 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 li ne progression is 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 t he content in the body area is presented for viewing with the page in landscape orientation and the con tent in the header and footer areas is presented for viewing with the page in portrait orientation. Thus f or body layout B, in the body area the character path and line progression is 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 is 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

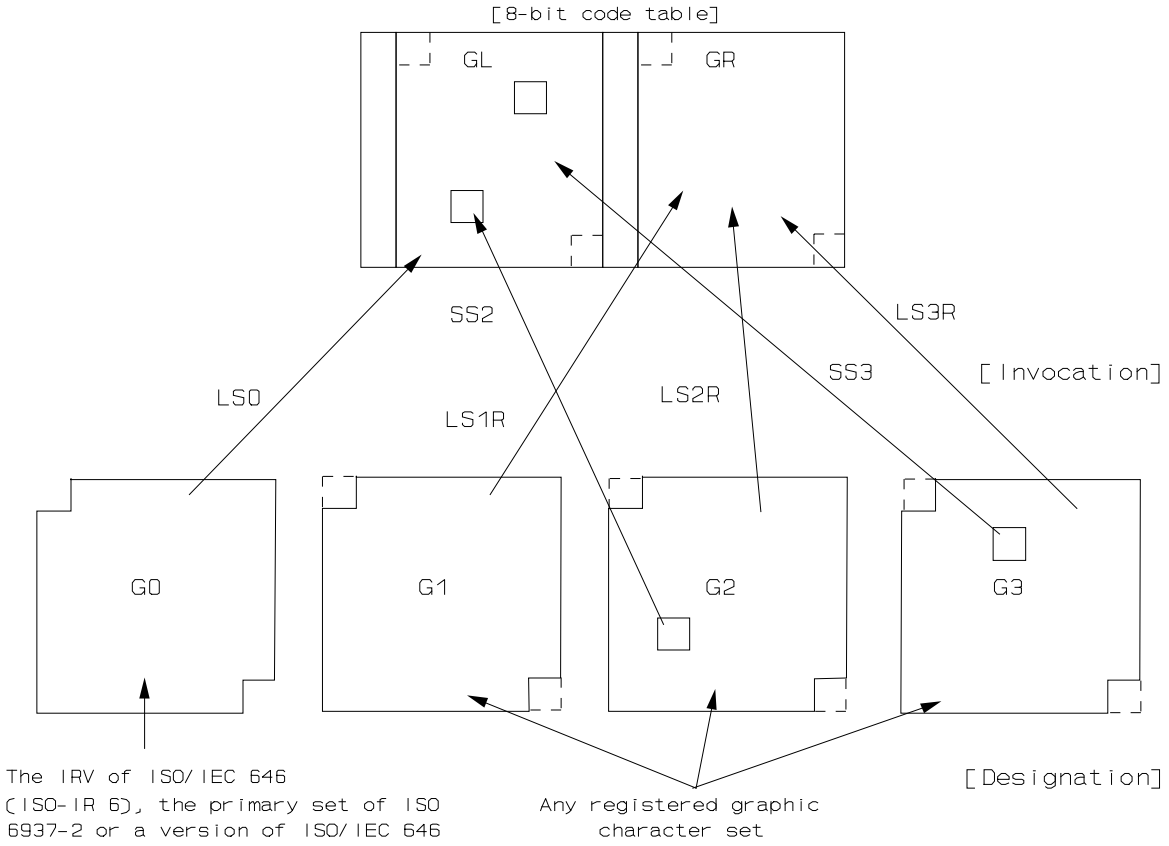

Figure 15 - Code extension features (all possible cases)

is allowed in these areas. Thus, in this case, the character path and line progression are specified as 2 70 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 t he presentation attributes "character path" and "line progression" respectively. These values cannot be changed within a content portion.

6.4.1.8 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.

6.4.1.9 Character orientation

The character orientation may be specified as 0 or 90 degrees, depending on whether vertical or horizo ntal 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 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.

6.4.1.10 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.

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, italicized/not italicized, underlined/doubly underlined/not underlined and crossed-out/not crossed-out. One mode from each mutually exclusive set may be in operation at any p oint 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 presentati on 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.

6.4.1.11 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 n ot 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 s et.

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 us ing the control function selective tabulation (STAB).

The tabulation reference numbers used in the control function STAB and associated presentation attribu te "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.Indentation
Indentation is the distance between the first character on a line of content and the position of the margin that is in the direction opposite to the direction of the character path. Thus the value of the indentation specified determines the line home position (as defined in CCITT Rec. T.416 | ISO 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.

6.4.1.12 Alignment

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

The following values of alignment may be specified as basic:

— start aligned;

— end aligned;

— centred;

— justified.

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

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

6.4.1.13 First line format

This feature specifies how the first line of the content associated with a basic component is to be laid o ut 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 iden tifier 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 | I SO 8613-6.

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

6.4.1.14 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 fra me or page when the content associated with a basic logical component is split over two frames or pag es. 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.

6.4.1.15 Fonts

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

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

The fonts that may be used within the content associated with each basic component are specified by t

he presentation attribute "character fonts". Up to 10 fonts taken from the list specified by the attribute "f onts 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 functi on select graphic rendition (SGR).

The document profile may specify, using the attribute "character fonts", a default set of up to 10 fonts th at are applicable to the whole document.Reverse character strings
Bi-directional writing is supported by this Specification. 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 reve rse 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.

6.4.1.16 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.

6.4.1.17 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 attribut es associated with the fonts used within the two lines and may vary from one line to the next. This pro cess is application dependent.

6.4.1.18 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.

6.4.1.19 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 oc cur respectively when the content is laid out.

6.4.1.20 Substitution of characters

The control function substitute (SUB) is provided to represent characters produced by a local system th at cannot be represented by a character within a character set supported by this Specification.

6.4.1.21 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.

6.4.1.22 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;

ST — start of string;

SOS — string terminator;

— code extension control functions (see 6.5.1.4).

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

6.4.1.23 Formatting the content

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

6.4.2 Raster graphics content

6.4.2.1 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 indicate d.

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

— type of coding;

— compression;

— pel spacing;

— spacing ratio;

— image dimensions.

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

6.4.2.2 Raster graphics content architecture

Only the formatted processable raster graphics content architecture class may be used in documents th at conform to this document application profile. This type of content may be used in processable, form atted 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 subseque nt 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.

Table 3 — List of number string identifiers for logical constituent constraints

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

6.4.2.3 Raster graphics encoding methods

The content may be encoded in accordance with the encoding schemes defined in CCITT Rec.s 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 f orms of encoding may be used in a single document, and all are basic. Uncompressed mode of encodi ng 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 th at 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 Specification places no

Table 4 — List of number string identifiers for layout constituent constraints

Layout constituent constraints	Numeric string identifier	Layout constituent constraints	Numeric string identifier
DocumentLayoutRoot	0	SourcedContentFixed	18
PageSet	1	SourcedContentVariable	19
Page	2	BasicHeader	27
RectoPage	3	BasicBody	28
VersoPage	4	GenericBlock	29
CompositeHeader	5	SpecificBlock	30
VariableCompositeBody	7	CompositeFooter	32
ColumnFixed	8	BasicFooter	33
ColumnVariable	9	TableHeader	34
SnakingColumns	10	TableArea	36
SynchronizedColumns	11	TableLabel	37
BasicFloat	12	RowArea	40
FootnoteArea	15	Cell	41
ArrangedContentFixed	16	TableLabelContent	44
ArrangedContentVariable	17		

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 architectu re class, then the default encoding method is assumed to be that defined in CCITT Recommendation T. 6.

6.4.2.4 Pel path and line progression

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

6.4.2.5 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 attribut e.

6.4.2.6 Pel spacing

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

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

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

NOTES

1 The basic pel spacing values listed above are equivalent to resolutions of 75, 100, 150, 200, 240, 300, 400, 600 and 1200 pels per 25,4 mm respectively when the document is imaged in its specified size.

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

6.4.2.7 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 t han 1 BMU. Also, the line spacing need not be an integral number of BMUs. All values are basic.

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

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

6.4.2.8 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 spaci ng" is 'null'.

6.4.3 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. Ea ch CGM figure may specify its minimum dimensions.

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

Miscellaneous features

6.5.1 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-document.

The constituent constraints in the resource-documentmay 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 attribute "resource": however, a generic-document used as a resource-document may contain any combination of generic constituent constraints which is conformant to this document application profile.

6.5.2 External-documents

In the case of processable and formatted processable, the generic logical structure and the generic layo ut 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.

6.5.3 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 "bor der". 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.

Border shall not be specified for the constituents GenericBlock and SpecificBlock.

6.5.4 Application comments

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

This attribute is structured so that it contains two fields. The first field is mandatory when the attribute is specified and contains a numeric string which uniquely identifies the constituent constraint applicable to the constituent for which the attribute is specified. This facilitates the processing of documents. A list of these identifiers is given in Tables 3 and 4.

NOTES

1 The values of the constituent constraint numeric identifiers are not unique between the logical and lay out 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 Recommendations | International Standardized Profiles to which this Specification 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 Specification and the interpretation of this field depe

nds 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.

6.5.5 Alternative representation

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

The specification and use of this attribute is optional. The string of characters specified shall belong to the character repertoires indicated in the document profile attribute "alternative representation character sets" (see 6.7.4.3). If the latter attribute is not explicitly specified in the document profile, then the default defined in CCITT Rec. T.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.

6.5.6 Automatic numbering and referencing mechanisms

6.5.6.1 Introduction

This Specification 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 fr om other parts of a document in order to support a general referencing mechanism within a document.

To achieve these features, this Specification 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 Specification or t he mechanisms used to achieve these numbering and referencing schemes.

6.5.6.2 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> se rves 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 zer oes.

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/IEC 646, ISO-IR 6, plus the character space. Any other character repertoire may be used provid

ed 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.

6.5.6.2.1 Binding 'prefixes-<n>'

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

6.5.6.2.2 Binding 'suffixes-<n>'

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

6.5.6.2.3 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.

6.5.6.2.4 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. Binding 'separator-<n>'
This binding specifies a character string that is typically used to represent the separators between num erals in a string represented by the binding 'numberstring-<n>'. An example is the string "3.4.3.6" where the character "." forms the separator.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 ot her part of the document.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. 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 FOD26 | PM-26.

6.5.6.2.5 Binding 'fnotestring'

This binding specifies a character string which is the equivalent of the number represented by the bindi ng 'fnotenumber'. This binding may only be specified for the logical constituent constraint Footnote, and is provided for compatibility with FOD26 | PM-26.

6.5.6.3 Numbering of segments

The constituent constraint Number contains a content generator which, when evaluated during the layo ut process, produces a character string that serves to identify the NumberedSegment to which the Num ber 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 'prefi x-<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 (that is, 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 a 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 succes sive 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 sha ll 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>'; a nd 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 tables and lists of items.

6.5.6.4 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.)

6.5.6.5 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.

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 a re 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.

6.5.6.6 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 fi elds <pre-str> and <suf-str> are optional prefix and suffix character strings respectively which may be of any length and are predefined in the expression.

The field <num-str> is 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.

6.5.6.7 Page numbering

The constituent constraint PageNumber is specifically provided to represent common content that contai ns 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, and are predefined in the expression.

The field <num-str> is the page number. This consists of a single number derived from the binding '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 low er case Roman numeric string or an equivalent upper or lower case alphabetic string.

The binding 'PGnum' is initialized at the document layout root or page set level (using the macro INITIALISEPGNUMBER 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 PGnum that is associated with the current page being laid out.

Similar numbering is applicable for page sets.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 a nd pages. Particular strings specified by 'string-<n>' may be referenced.

This constituent constraint contains a content generator which, when evaluated, produces a character st ring 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 are predefined in the expression.

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 o n 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.

6.5.6.8 Common references

CommonReference is a constituent constraint within the common part of the logical structure of a docu ment 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. Exampl es 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 t he 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 sea rch for the required binding is made on the layout objects superior to the referenced pa ge 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.

6.5.7 User readable comments

Information which is to be interpreted as comments relevant to constituents and associated content port ions 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 Rec. 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.

6.5.8 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 a ssist 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 Rec. 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 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.

6.6.1 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.

6.6.1.1 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 necess ary to indicate that these structures are present and form part of the document.

6.6.1.2 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 t he document profile.

6.6.1.3 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.

6.6.2 Document identification information

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

6.6.2.1 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".

6.6.2.2 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".

6.6.2.3 Content architecture classes information

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

6.6.2.4 ODA version information

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

6.6.2.5 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".

6.6.3 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 va lue specified in CCITT Rec. 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.

6.6.4 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 FO D26 | PM-26.

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

6.6.4.1 Profile character sets

Some document profile attributes have values consisting of character strings, for example, the document management attributes. The character sets used in these character strings are specified by the docum ent 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 an d G1 sets are implicitly invoked by this code extension announcer.

— G0 set: only ISO-IR 6 (the IRV of ISO/IEC 646), ISO-IR 2 (the primary set of ISO 6937-2), 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 Rec. T.410 seri es | ISO 8613 is assumed.

6.6.4.2 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 nam e" (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 with out 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 t his 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 Rec. T.410 series | ISO 8613 is assumed.

6.6.4.3 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 explic itly specified in the document profile, the default defined in CCITT Rec. T.410 series | ISO 8613 is assumed.

6.6.5 Fonts list

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

6.6.6 Document management attributes

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

— document description (see the note below);

— dates and times;

— originators;

— other user information;

— external references;

— local file references;

— content attributes;

— security information.

The attributes that may be used to specify this information are defined in CCITT Rec. T.414 | ISO 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 n ot 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 cann ot be changed in the document management attributes.

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

6.6.6.1 Additional information

This attribute is required to be specified with the value "Level 2 to Level 3 Migration DAP".

This permits applications to distinguish FOD36 | PM-36 documents which adhere to the Implementation Support Requirements specified in this document from fully-functional FOD36 | PM-36 documents.

7 Specification of constituent constraints

This clause specifies the definition of the constituent constraints which may be represented by data stre ams conforming to this Specification.

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, a nd define the allowed structures that are supported by this Specification.

The structure diagrams illustrating the layout structures are shown in 7.1.2. The macros indicated in the se diagrams are defined in 7.4.1.

7.1.1 Diagrams of relationships of logical constituents

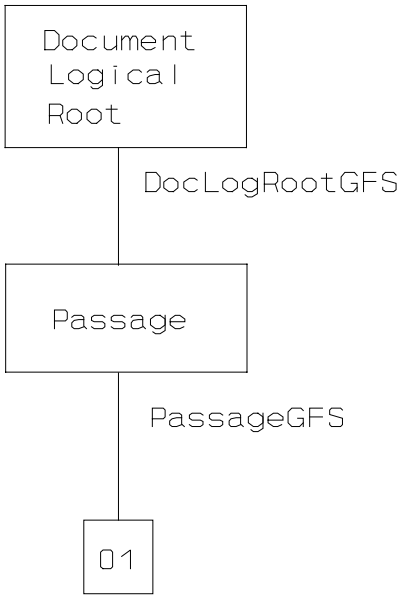

Figure 16 — DocumentLogicalRoot, 1st level

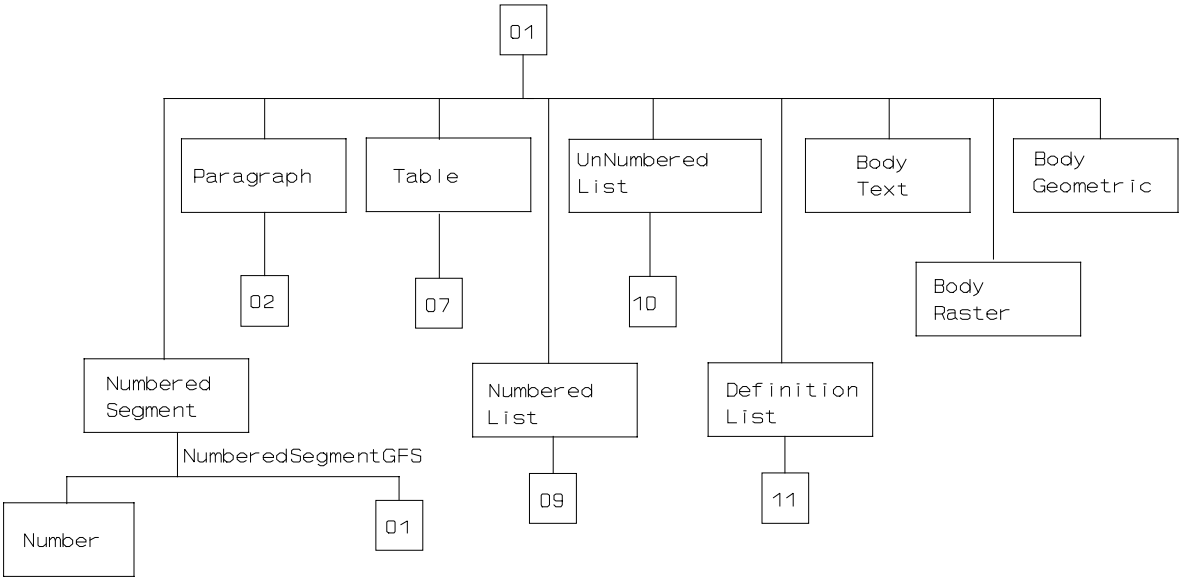

Figure 17 — DocumentLogicalRoot, 2nd level

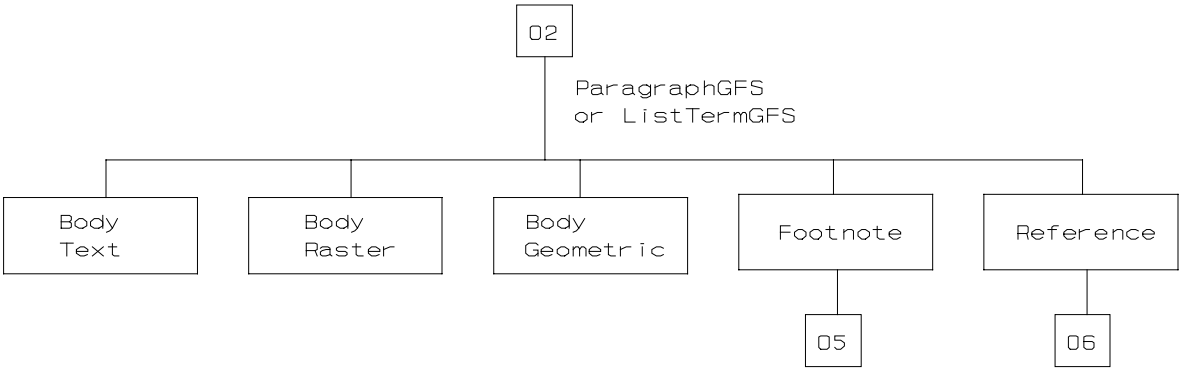

Figure 18 — Paragraph and ListTerm

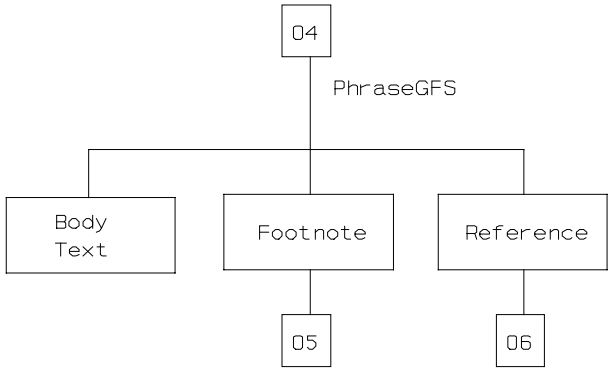

Figure 19 — Phrase

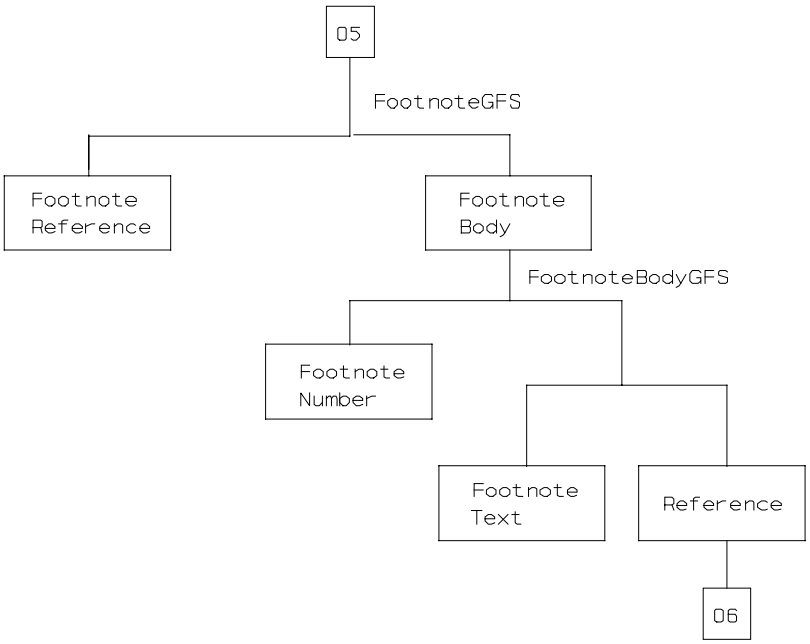

Figure 20 — Footnote

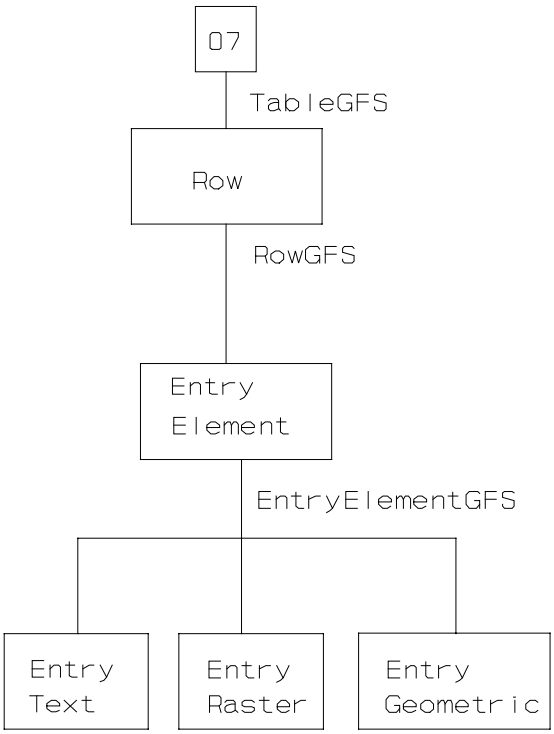

Figure 21 — Table

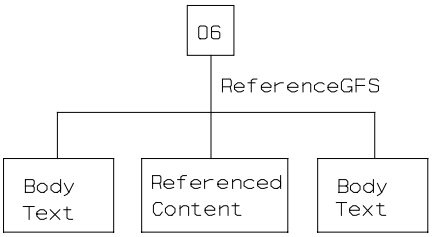

Figure 22 — Reference

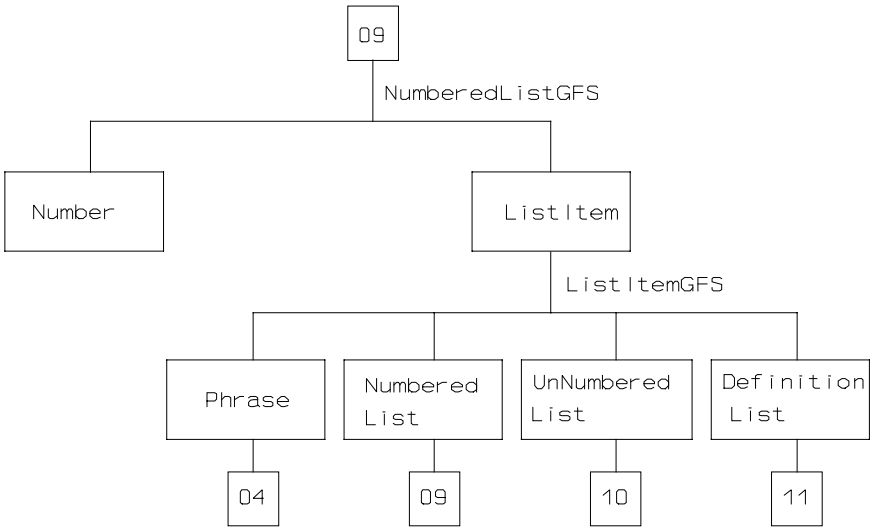

Figure 23 — NumberedList

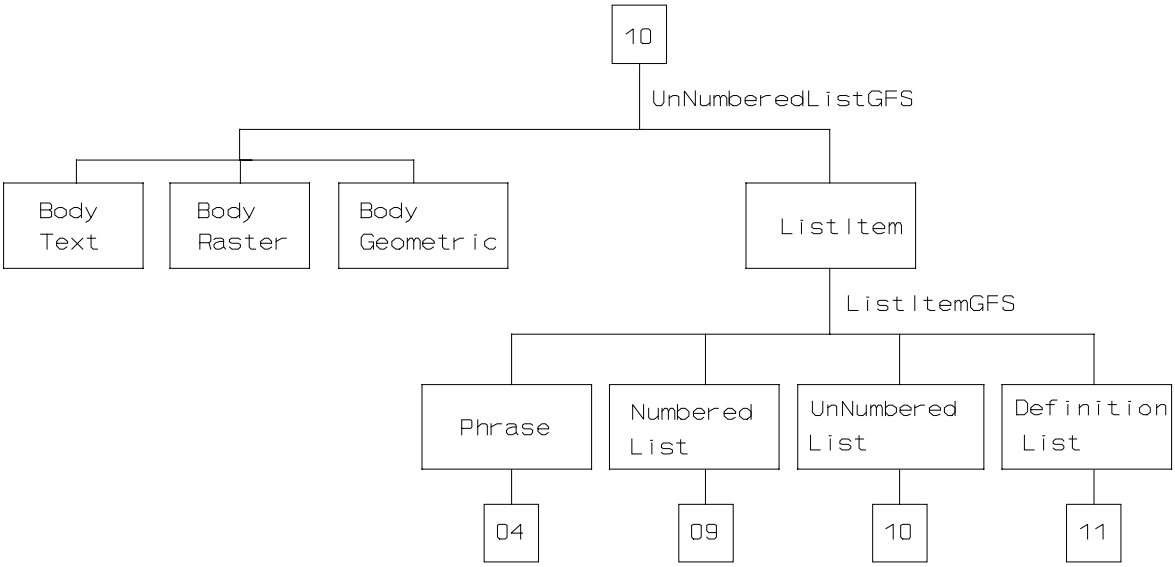

Figure 24 — UnNumberedList

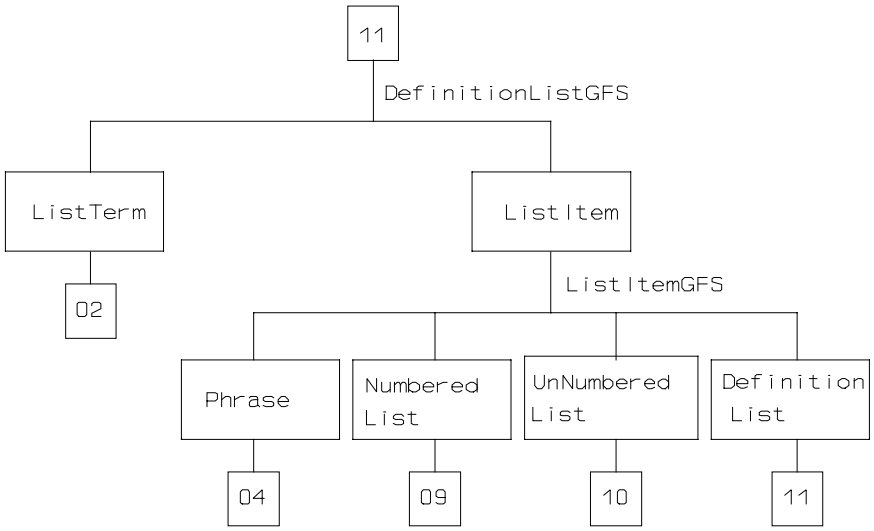

Figure 25 — DefinitionList

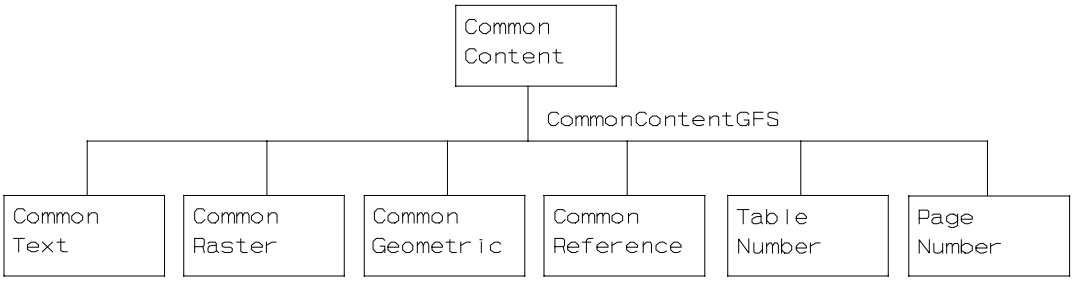

Figure 26 — CommonContent

Diagrams of relationships of layout constituents

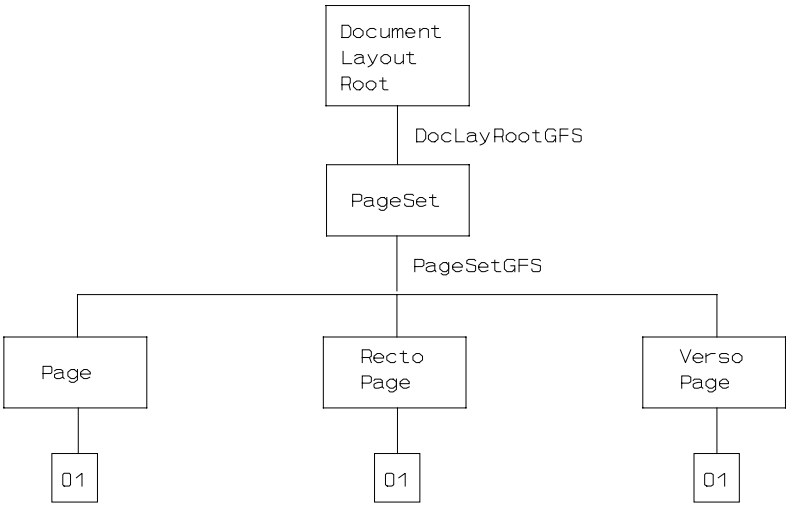

Figure 27 — DocumentLayoutRoot

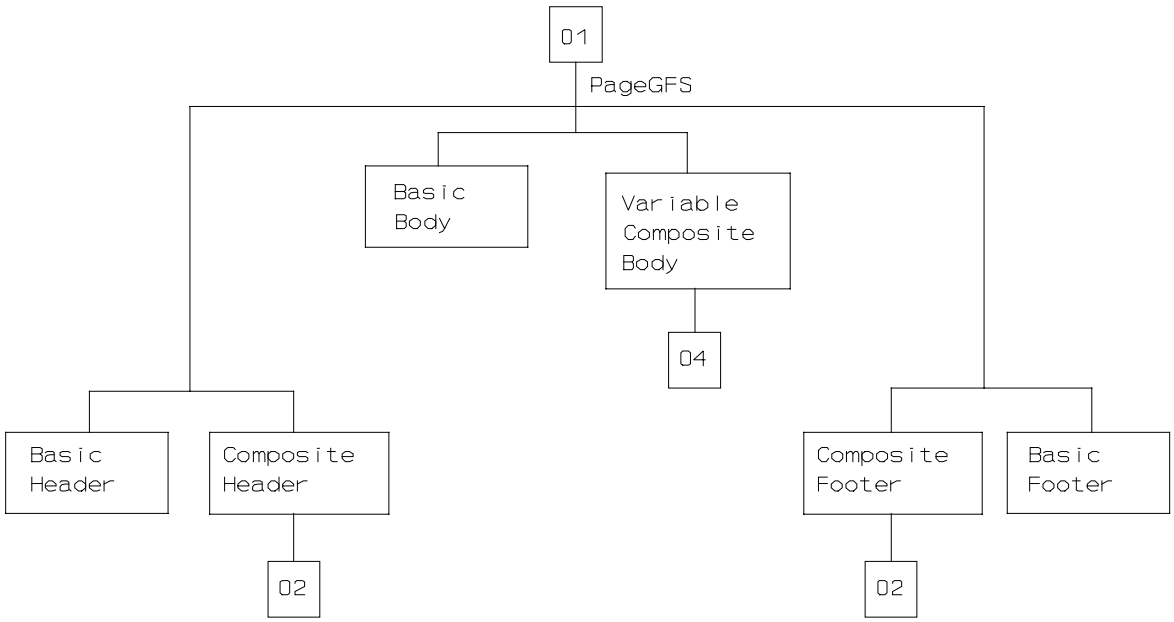

Figure 28 — Page, RectoPage and VersoPage

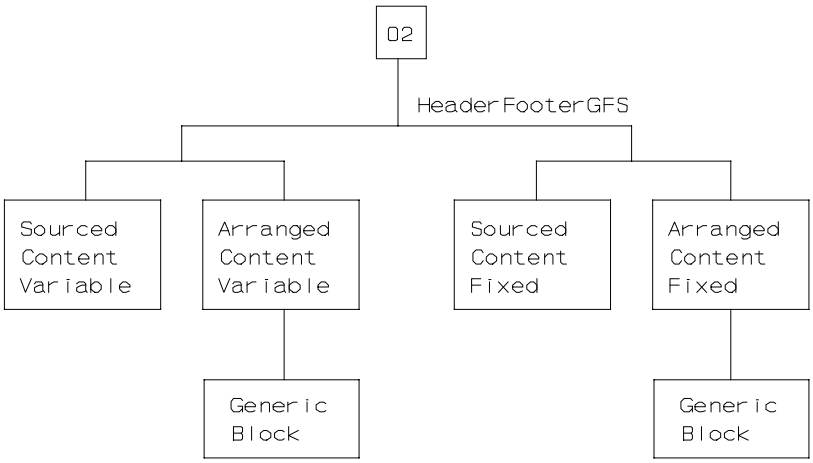

Figure 29 — CompositeHeader and CompositeFooter

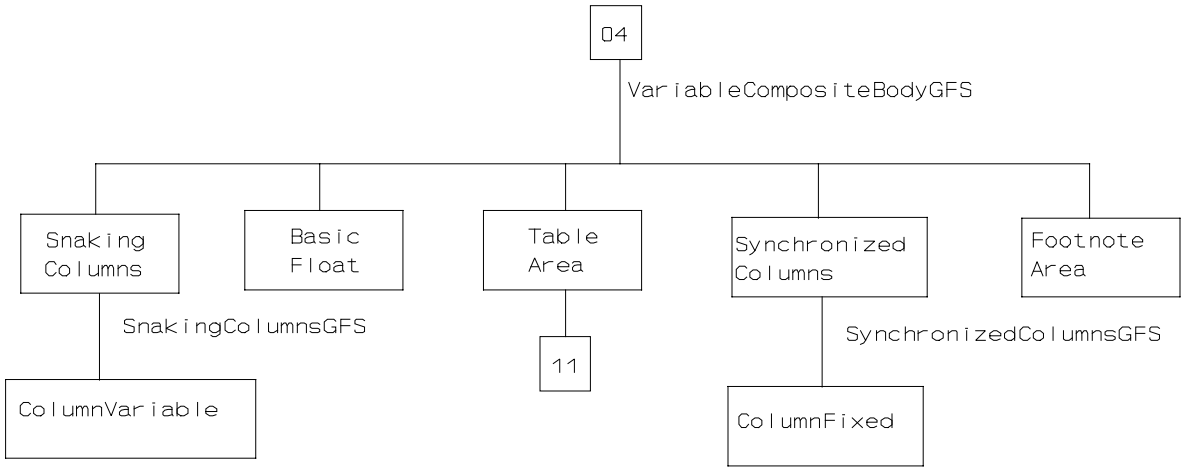

Figure 30 — VariableCompositeBody

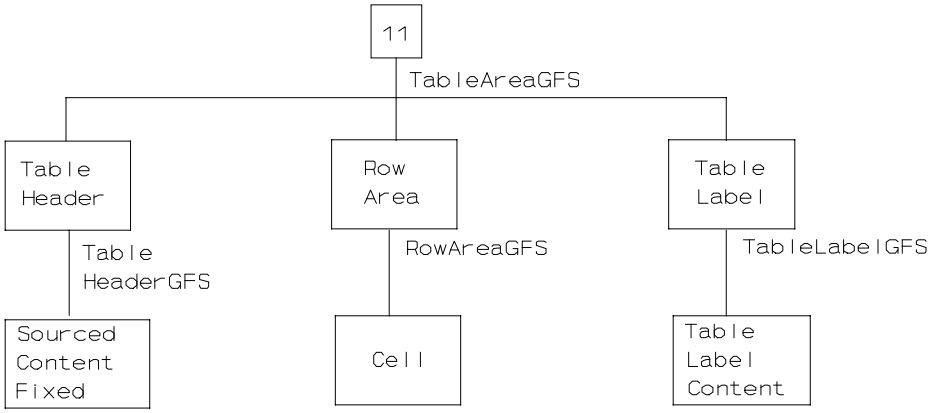

Figure 31 — TableArea

7.1.2 Notation

This clause is written in accordance with the Document Application Profile Proforma and Notation (DAP PN) of CCITT Rec. T.411 | ISO 8613-1, Annex F. The following clarifications and minor extensions appl y:

a) [Clarification]
The value range definition for the attributes "subordinates" and "imaging-order" specify t he set of object instances that may occur. The ordering and number (which may be zer o) of object instances for the attribute "subordinates" shall be in accordance with the val ue 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 gra phic characters.

c) [Extension]
In order to write the specification of the usage of character sets and code extension con trol functions precisely, the following extensions are applied:

1) Table 5 defines the symbols that are introduced to denote shift functions.

Table 5 — Symbols to denote shift functions

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

2) <escape-sequence> is extended to include shift functions:
<escape-sequence> ::= 'ESC' <octet>... [<invocation-control-function>];
<invocation-control-function> ::= 'LS0'|'LS1R'|'LS2R'|'LS3R'|'SS2'|'SS3';

3) Data type specification for #ESC in content information is extended as:
<escape-sequence>...

[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.

d) [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.

e) [Extension]
"|" is used in CASE SUPERIOR expressions in the following format in order to shorten the text:
CASE SUPERIOR ({const1|const2|...|constn}(aaaa)) OF {.....}
where "const1, const2, ... , constn" are names of constituent constraints, and "aaaa" is the name of an attribute.

This expression is equivalent to the following expression:
CASE SUPERIOR (const1(aaaa)) OF {.....}
CASE SUPERIOR (const2(aaaa)) OF {.....}
. . . .
CASE SUPERIOR (constn(aaaa)) OF {.....}

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

7.2 Document profile constituent constraints

7.2.1 Macro definitions

```
DEFINE(FC, "ASN.1{2 8 2 6 0}" -- formatted character content --)
DEFINE(PC, "ASN.1{2 8 2 6 1}" -- processable character content --)
DEFINE(FPC, "ASN.1{2 8 2 6 2}" -- formatted processable character content --)
DEFINE(FPR, "ASN.1{2 8 2 7 2}" -- formatted processable raster graphics content --)
DEFINE(FPG, "ASN.1{2 8 2 8 0}" -- formatted processable geometric graphics content --)


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

DEFINE(DAC, "DocumentProfile (Document-architecture-class)")
DEFINE(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{<=14030}},
                 REQ #vertical-dimension
                        {REQ #fixed-dimension {12401..19840}}
               |REQ #horizontal-dimension
               {REQ #fixed-dimension{9241..14030}},
                 REQ #vertical-dimension
                        {REQ #fixed-dimension {<=19840}}
                                -- up to ISO A3 portrait --
               |REQ #horizontal-dimension
                        {REQ #fixed-dimension {<=19840}},
                 REQ #vertical-dimension
                        {REQ #fixed-dimension {9241..14030}}
               |REQ #horizontal-dimension
                        {REQ #fixed-dimension {12401..19840}},
                 REQ #vertical-dimension
                        {REQ #fixed-dimension {<=14030}}
                                -- up to ISO A3 landscape --
               |REQ #horizontal-dimension
```

```
                    {REQ #fixed-dimension {<=13200}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {12401..20400}}
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {9241..13200}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {<=20400}}
                                -- up to ANSI B portrait --
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {<=20400}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {9241..13200}}
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {12401..20400}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {<=13200}}
                                -- up to ANSI B 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 {<=14030}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {<=19840}} -- up to ISO A3 portrait --
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {<=19840}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {<=14030}} -- up to ISO A3 landscape --
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {<=13200}},
                REQ #vertical-dimension
                    {REQ #fixed-dimension {<=20400}} -- up to ANSI B portrait --
               |REQ #horizontal-dimension
                    {REQ #fixed-dimension {<=20400}},
                REQ #vertical-dimension
```

```
                    {REQ #fixed-dimension {<=13200}} -- up to ANSI B landscape --
")


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 {10200},   REQ #vertical-dimension {16800}
                                             -- ANSI legal portrait --
      | REQ #horizontal-dimension {16800},   REQ #vertical-dimension {10200}
                                             -- ANSI legal landscape --
      | REQ #horizontal-dimension {10200},   REQ #vertical-dimension {13200}
                                             -- ANSI A portrait  --
      | REQ #horizontal-dimension {13200},   REQ #vertical-dimension {10200}
                                             -- ANSI A landscape --
      | REQ #horizontal-dimension {13200},   REQ #vertical-dimension {20400}
                                             -- ANSI B portrait --
      | REQ #horizontal-dimension {20400},   REQ #vertical-dimension {13200}
                                             -- ANSI B landscape --
      | REQ #horizontal-dimension {12141},   REQ #vertical-dimension {17196}
                                             -- 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'
        |'italicised'|'underlined'|'crossed-out'
        |'primary-font'|'first-alternative-font'
        |'second-alternative-font' |'third-alternative-font'
        |'fourth-alternative-font' |'fifth-alternative-font'
        |'sixth-alternative-font'|'seventh-alternative-font'
        |'eighth-alternative-font' |'ninth-alternative-font'
        |'doubly-underlined' |'normal-intensity' |'not-italicised'
        |'not-underlined'|'not-crossed-out' }... ")


--      Macro defining permissible code extension announcers. 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,

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

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

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

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

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

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


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

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


-- 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 t
        he 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  #layout-path              {'0-degrees'|'180-degrees'|'270-degrees'},

    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            {'0-degrees'|'90-degrees'},
            PERM  #code-extension-announcers        {$DAP-DEFAULT-CDEXTEN},
            PERM  #first-line-offset                {ANY_VALUE},
            PERM  #graphic-character-sets           {$DAP-DEFAULT-GRCHAR},
            PERM  #graphic-character-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  #indentation                      {ANY_VALUE},
            PERM  #orphan-size                      {ANY_VALUE},
            PERM  #proportional-line-spacing        {ANY_VALUE},
            PERM  #widow-size                       {ANY_VALUE},
            PERM  #initial-offset                   {ANY_VALUE}
      },

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

  },

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

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

```
REQ   Interchange-format-class      {-- See clause 8 for the definition of the permitted values for
                                    this attribute. --},
REQ   Oda-version                   {REQ #standard-or-recommendation
                                        {"CCITT Rec. T.410 series (1988) | ISO 8613 (1989); version 1.1"},
                                     REQ #publication-date          {"1992-01-01"}},


      -- non basic document characteristics --

PERM  Profile-character-sets                          {$PROFCHAR},

PERM  Comments-character-sets                         {$COMCHAR},

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

PERM  Page-dimensions     {PMUL{$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 | PM-26 --

PERM  Borders             {ANY_VALUE},
      -- Any values need not be declared, they may be specified here for upwards compatibility
      from FOD26 | PM-26 --

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 | PM-26 --

      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 | PM-26 --

      PMUL {PERM #character-spacing         {<100 | 100 | 160 | 200}},
            -- These values need not be declared, they may be specified here for upwards
            compatibility from FOD26 | PM-26 --

      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 | PM-26 --

      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 | PM-26 --

        PERM #line-progression                          {'90-degrees'},
            -- This value need not be declared, they may be specified here for upwards compatibility
            from FOD26 | PM-26 --

        PMUL {PERM #graphic-character-sets          {$NON-BASIC-GRCHAR}},
        PMUL {PERM #graphic-character-subrepertoire                 {ANY_VALUE}}
    },

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

        PMUL {PERM #spacing-ratio
                {REQ #line-spacing-value     {ANY_VALUE} EXCEPT {1},
                 REQ #pel-spacing-value      {ANY_VALUE} EXCEPT {1}}}
     }
  },

        -- additional document characteristics --

  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},
    REQ   Additional-information             {"Level 2 to Level 3 Migration DAP"},

        -- 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(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(BodyText)
                                  | OBJECT_ID_OF(Reference)
                                  | OBJECT_ID_OF(ReferencedContent)
                                  | OBJECT_ID_OF(BodyRaster)
                                  | OBJECT_ID_OF(BodyGeometric)
                                  | 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(EntryElement)
                                  | 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(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(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(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(EntryElement)
                                        | 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 a segment number binding.  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 a segment number binding.  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, table 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 val
        ue such as in segment 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, table number, list number, footnote number or page numbe
        rs. 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.

DEFINE(FOOTNOTECATEGORY, "
 <footnotecategory>      ::=      ""Footnote"";
")

--      NOTE: The following macro is called INITIALISEANY in FOD26 | PM26. --

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         {>=0}
         |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 initialise/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 initialise fnotenumber and fnotestring bindings. --

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

```
")


-- Used to increment fnotenumber binding. --

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


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

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


--      Used to reset the footnote number string to a string literal. This provides a mechanism for settin
        g 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: The following macro is called INITIALISEPGNUM in FOD26 | PM26.
        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 us
        ed for the compatibility with FOD26 | PM-26. --

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


--      Used to increment PGnum binding.  --
--      NOTE: The following macro is called PAGENUMBER in FOD26 | PM26. --

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 te
        xt for segment 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 gener
        ate 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>       ::=      ANY_STRING;
 <suf-str>       ::=      ANY_STRING;
 $NUMBERSTRING
")


--      This string expression is allowed in a content generator for PageNumber to automatically gener
        ate text for a page number.  --
--      NOTE: The following macro is called HEADERFOOTERSTRING in FOD26 | PM26.
        A page number may be generated either from <number> or <numberstring> or from PGnum.
        PGnum is kept for compatibility with FOD26 | PM-26. --

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

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

 <numeric-expr>    ::= B_REF(SUP(<layout-object-1>))(""PGnum"")
                   | B_REF(<layout-object-2>)(""PGnum"");

 <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);
")
```

-- This string expression is allowed in a content generator for FootnoteNumber and FootnoteReference to automatically generate text for a footnote number. --
-- NOTE: The following macro is called FNOTENUMBER in FOD26 | PM-26. --

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

-- This string expression is allowed in a content generator for ReferencedContent to automatically generate text for references such as to segment numbers, table numbers, list numbers, footnot e numbers and <string> bindings associated with a referring (i.e. a target) logical object, or to p age numbers, pageset numbers and <string> bindings associated with a layout object in which t he 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. '(Se e' and ')'. --

```
 <pre-str>      ::= ANY_STRING;
 <suf-str>      ::= 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> ::= <ref-str-b>;
 <ref-str-b> ::= B_REF(SUP(<target-logical-object-1>))(""fnotestring"")
                | B_REF(<target-logical-object>)(""fnotestring"");

 <ref-pgnum> ::= <ref-str-c>;
 <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> ::= <ref-str-e>;
<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, list numbers, footnot e 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>       ::= ANY_STRING;
 <suf-str>       ::= 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> ::=  <ref-str-b>;
 <ref-str-b> ::= B_REF(SUP(<current-logical-object>))(""fnotestring"")
                 | B_REF(<current-logical-object>)(""fnotestring"");

 <ref-pgnum> ::= <ref-str-c>;
 <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> ::= <ref-str-e>;
 <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
")

DEFINE(DocLogRootGFS, "
 <construction-expr>    ::=     <construction-term>
                                | <construction-type>;

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

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

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


DEFINE(CONSTRAINT-1,  "
```

```
 <constraint-1>          ::=     <construction-term>
                                         | <construction-type>;

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

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

 <construction-factor>   ::=     OBJECT_CLASS_ID_OF(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(Table)
                                         | OBJECT_CLASS_ID_OF(NumberedList)
                                         | OBJECT_CLASS_ID_OF(UnNumberedList)
                                         | OBJECT_CLASS_ID_OF(DefinitionList)
                                         | <construction-type>;
")


DEFINE(PassageGFS, "
 <construction-expr>     ::=     <constraint-1>;

 $CONSTRAINT-1
")


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(Reference)
                                        | <construction-type>;
")


DEFINE(ParagraphGFS, "$CONSTRAINT-3")


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(Footnote)
                                        | OBJECT_CLASS_ID_OF(Reference)
                                        | OBJECT_CLASS_ID_OF(BodyText)
                                        | <construction-type>;
")


DEFINE(PhraseGFS,   "$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(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>;
 <simple-table>          ::=     REP OBJECT_CLASS_ID_OF(EntryElement)
                                 | REP CHO(OBJECT_CLASS_ID_OF(EntryElement)...)
                                 | SEQ(OBJECT_CLASS_ID_OF(EntryElement)...);
")


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:
   REQ    Object-type                {VIRTUAL},
   REQ    Object-class-identifier    {ANY_VALUE}
SPECIFIC:
   PERM   Object-type                {VIRTUAL},
   REQ    Object-identifier          {ANY_VALUE},
   REQ    Object-class               {VIRTUAL}
SPECIFIC_AND_GENERIC:
   PERM   User-readable-comments     {ANY_STRING},
   PERM   User-visible-name          {ANY_STRING}
}
```

7.3.2.2 FACTOR COMP-LOGICAL

```
:ANY-LOGICAL {
GENERIC:
   REQ    Object-type                {'composite-logical-object'}
SPECIFIC:
   REQ    Subordinates               {VIRTUAL},
   PERM   Object-type                {'composite-logical-object'}
}
```

7.3.2.3 FACTOR BASIC-LOGICAL

```
:ANY-LOGICAL {
GENERIC:
   REQ    Object-type                {'basic-logical-object'},
SPECIFIC:
   PERM   Object-type                {'basic-logical-object'}
}
```

7.3.2.4 FACTOR ANY-COMMON

```
{
GENERIC:
   REQ    Object-type                {VIRTUAL},
   REQ    Object-class-identifier    {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:
   REQ    Object-type                {'document-logical-root'},
   REQ    Generator-for-subordinates {$DocLogRootGFS},
   REQ    Application-comments       {REQ #constraint-name {"0"},
                                          PERM #external-data {ANY_VALUE}}
SPECIFIC:
   PERM   Object-type                {'document-logical-root'},
   REQ    Object-class               {OBJECT_CLASS_ID_OF(DocumentLogicalRoot)},
   REQ    Subordinates               {SUB_ID_OF(Passage)+},
   PERM   Application-comments       {REQ #constraint-name {"0"},
                                       PERM #external-data  {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Bindings                   {PMUL {$INITIALISEBINDINGS},
                                       PERM {$INITIALISEFNOTE}},
}
```

7.3.3.2 Passage

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$PassageGFS},
   REQ    Application-comments         {REQ #constraint-name {"1"},
                                        PERM #external-data  {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(Passage)},
   REQ    Subordinates                 {SUB_ID_OF(NumberedSegment)+,
                                        SUB_ID_OF(BodyText)+,
                                        SUB_ID_OF(BodyRaster)+,
                                        SUB_ID_OF(BodyGeometric)+,
                                        SUB_ID_OF(Paragraph)+,
                                        SUB_ID_OF(Table)+,
                                        SUB_ID_OF(NumberedList)+,
                                        SUB_ID_OF(DefinitionList)+,
                                        SUB_ID_OF(UnNumberedList)+},
   PERM   Application-comments         {REQ  #constraint-name {"1"},
                                        PERM #external-data   {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Bindings                     {PMUL {$INITIALISEBINDINGS},
                                        PERM {$INITIALISEFNOTE}},
   PERM   Layout-style                 {STYLE_ID_OF(LStyle1)}
}
```

7.3.3.3 NumberedSegment

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$NumberedSegmentGFS},
   REQ    Application-comments         {REQ #constraint-name {"2"},
                                        PERM #external-data {ANY_VALUE}},
   REQ    Bindings                     {PMUL {$SPECIFYBINDINGS}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(NumberedSegment)},
   REQ    Subordinates                 {SUB_ID_OF(Number),
                                       SUB_ID_OF(NumberedSegment)+,
                                       SUB_ID_OF(BodyText)+,
                                       SUB_ID_OF(BodyRaster)+,
                                       SUB_ID_OF(BodyGeometric)+,
                                       SUB_ID_OF(Paragraph)+,
                                       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}},
   PERM   Bindings                     {PMUL {$INITIALISEBINDINGS}}
SPECIFIC_AND_GENERIC:
   PERM   Layout-style                 {STYLE_ID_OF(LStyle4)}
}
```

7.3.3.4 Number

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-generator            {$SEGMENTNUMBER},
   REQ    Application-comments         {REQ  #constraint-name {"3"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(Number)},
   PERM   Application-comments         {REQ #constraint-name {"3"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Layout-style                 {STYLE_ID_OF(LStyle2)},
   PERM   Presentation-style           {STYLE_ID_OF(PStyle1)},
   PERM   Content-architecture-class   {$FC|$PC|$FPC}
}
```

7.3.3.5 Paragraph

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$ParagraphGFS},
   REQ    Application-comments         {REQ #constraint-name {"6"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(Paragraph)},
   REQ    Subordinates                 {SUB_ID_OF(BodyText)+,
                                        SUB_ID_OF(Footnote)+,
                                        SUB_ID_OF(BodyRaster)+,
                                        SUB_ID_OF(BodyGeometric)+,
                                        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(LStyle4)}
}
```

7.3.3.6 Phrase

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$PhraseGFS},
   REQ    Application-comments         {REQ #constraint-name {"7"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(Phrase)},
   REQ    Subordinates                 {SUB_ID_OF(BodyText)+,
                                        SUB_ID_OF(Footnote)+,
                                        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(LStyle4)}
}
```

7.3.3.7 Footnote

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$FootnoteGFS},
   PERM   Bindings                    {REQ{$INCFNOTENUMBER,$FNOTENUMBERSTRING}
                                          | $FNOTESTRINGLITERAL},
   REQ    Application-comments        {REQ  #constraint-name {"8"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(Footnote)},
   REQ    Subordinates                {SUB_ID_OF(FootnoteReference),
                                       SUB_ID_OF(FootnoteBody)},
   PERM   Bindings                    {$FNOTESTRINGLITERAL}
                                       PERM {$FNOTESTRINGLITERAL}},
   PERM   Application-comments        {REQ  #constraint-name {"8"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Layout-style                {STYLE_ID_OF(LStyle4)}
}
```

7.3.3.8 FootnoteNumber

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-generator           {$FNNUMBER},
   REQ    Application-comments        {REQ #constraint-name {"9"},
                                       PERM #external-data {ANY_VALUE}}
   REQ    Layout-style                {STYLE_ID_OF(LStyle9)},
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(FootnoteNumber)},
   PERM   Application-comments        {REQ #constraint-name {"9"},
                                       PERM #external-data {ANY_VALUE}}
   PERM   Layout-style                {STYLE_ID_OF(LStyle9)},
SPECIFIC_AND_GENERIC:
   PERM   Presentation-style          {STYLE_ID_OF(PStyle1)},
   PERM   Content-architecture-class  {$FC|$PC|$FPC}
}
```

7.3.3.9 FootnoteReference

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-generator           {$FNNUMBER},
   REQ    Application-comments        {REQ #constraint-name {"10"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(FootnoteReference)},
```

```
    PERM    Application-comments        {REQ #constraint-name {"10"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM    Layout-style                {STYLE_ID_OF(LStyle10)},
    PERM    Presentation-style          {STYLE_ID_OF(PStyle1)},
    PERM    Content-architecture-class  {$FC|$PC|$FPC}
}
```

7.3.3.10 FootnoteBody

```
:COMP-LOGICAL {
GENERIC:
    REQ     Generator-for-subordinates  {$FootnoteBodyGFS},
    REQ     Application-comments        {REQ #constraint-name {"11"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    REQ     Object-class                {OBJECT_CLASS_ID_OF(FootnoteBody)},
    REQ     Subordinates                {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)}
}
```

7.3.3.11 FootnoteText

```
:BASIC-LOGICAL {
GENERIC:
    REQ     Application-comments        {REQ #constraint-name {"12"},
                                         PERM #external-data {ANY_VALUE}}
    REQ     Layout-style                {STYLE_ID_OF(LStyle6)},
SPECIFIC:
    REQ     Object-class                {OBJECT_CLASS_ID_OF(FootnoteText)},
    PERM    Application-comments        {REQ #constraint-name {"12"},
                                         PERM #external-data {ANY_VALUE}}
    PERM    Layout-style                {STYLE_ID_OF(LStyle6)},
    REQ     Content-portions            {CONTENT_ID_OF(Character-content-portion)+}
SPECIFIC_AND_GENERIC:
    PERM    Presentation-style          {STYLE_ID_OF(PStyle1)},
    PERM    Content-architecture-class  {$FC|$PC|$FPC},
}
```

7.3.3.12 BodyText

```
:BASIC-LOGICAL {
GENERIC:
    REQ     Application-comments        {REQ #constraint-name {"14"},
                                         PERM #external-data {ANY_VALUE}},
    PERM    Resource                            {ANY_VALUE}
```

```
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(BodyText)},
   PERM   Application-comments        {REQ #constraint-name {"14"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Layout-style                {STYLE_ID_OF(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. --
}
```

7.3.3.13 Reference

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$ReferenceGFS},
   REQ    Application-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(LStyle4)}
```

7.3.3.14 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}
}
```

7.3.3.15 BodyRaster

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-architecture-class    {$FPR},
   PERM   Resource                      {ANY_VALUE},
   REQ    Application-comments          {REQ #constraint-name {"17"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                  {OBJECT_CLASS_ID_OF(BodyRaster)},
   PERM   Content-architecture-class    {$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. --
}
```

7.3.3.16 BodyGeometric

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-architecture-class    {$FPG},
   PERM   Resource                      {ANY_VALUE},
   REQ    Application-comments          {REQ #constraint-name {"18"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                  {OBJECT_CLASS_ID_OF(BodyGeometric)},
   PERM   Content-architecture-class    {$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. --
}
```

7.3.3.17 CommonContent

```
:ANY-COMMON {
GENERIC:
   REQ    Object-type                   {'composite-logical-object'},
   REQ    Generator-for-subordinates    {$CommonContentGFS},
   REQ    Application-comments          {REQ #constraint-name {"19"},
                                         PERM #external-data {ANY_VALUE}},
```

7.3.3.18 CommonText

```
:ANY-COMMON {
GENERIC:
   REQ    Object-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},
   REQ    Application-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. --
}
```

7.3.3.19 CommonReference

```
:ANY-COMMON {
GENERIC:
   REQ    Object-type                  {'basic-logical-object'},
   REQ    Content-generator            {$COMMONREF},
   PERM   Layout-style                 {STYLE_ID_OF(LStyle3)},
   PERM   Presentation-style           {STYLE_ID_OF(PStyle4)},
   PERM   Content-architecture-class   {$FC|$PC|$FPC},
   REQ    Application-comments         {REQ #constraint-name {"37"},
                                        PERM #external-data {ANY_VALUE}}
}
```

7.3.3.20 CommonRaster

```
:ANY-COMMON {
GENERIC:
   REQ    Object-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)},
   REQ    Content-architecture-class   {$FPR},
   REQ    Application-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. --
}
```

7.3.3.21 CommonGeometric

```
:ANY-COMMON {
GENERIC:
   REQ    Object-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)},
    REQ    Content-architecture-class  {$FPG},
    REQ    Application-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. --
}
```

7.3.3.22 PageNumber

```
:ANY-COMMON {
GENERIC:
    REQ    Object-type                 {'basic-logical-object'},
    REQ    Content-generator           {$PGNUMBER},
    PERM   Layout-style                {STYLE_ID_OF(LStyle3)},
    PERM   Presentation-style          {STYLE_ID_OF(PStyle4)},
    PERM   Content-architecture-class  {$FC|$PC|$FPC},
    REQ    Application-comments        {REQ #constraint-name {"40"},
                                        PERM #external-data {ANY_VALUE}}
}
```

7.3.3.23 TableNumber

```
:ANY-COMMON {
GENERIC:
    REQ    Object-type                 {'basic-logical-object'},
    REQ    Content-generator           {$TABLENUMBER},
    PERM   Layout-style                {STYLE_ID_OF(LStyle3)},
    PERM   Presentation-style          {STYLE_ID_OF(PStyle4)},
    PERM   Content-architecture-class  {$FC|$PC|$FPC},
    REQ    Application-comments        {REQ #constraint-name {"44"},
                                        PERM #external-data {ANY_VALUE}}
}
```

7.3.3.24 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(LStyle4)}
}
```

7.3.3.25 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(LStyle4)}
}
```

7.3.3.26 DefinitionList

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$DefinitionListGFS},
   REQ    Application-comments         {REQ  #constraint-name {"27"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(DefinitionList)},
   REQ    Subordinates                 {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(LStyle4)}
}
```

7.3.3.27 ListItem

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates   {$ListItemGFS},
   REQ    Application-comments         {REQ #constraint-name {"28"},
                                        PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                 {OBJECT_CLASS_ID_OF(ListItem)},
   REQ    Subordinates                 {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(LStyle4)}
}
```

7.3.3.28 ListTerm

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$ListTermGFS},
   REQ    Application-comments        {REQ  #constraint-name {"29"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(ListTerm)},
   REQ    Subordinates                {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(LStyle4)}
}
```

7.3.3.29 Table

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$TableGFS},
   REQ    Application-comments        {REQ  #constraint-name {"30"},
                                       PERM #external-data {ANY_VALUE}},
   REQ    Layout-style                {STYLE_ID_OF(LStyleT4)}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(Table)},
   REQ    Subordinates                {SUB_ID_OF(Row)+},
   PERM   Application-comments        {REQ  #constraint-name {"30"},
                                       PERM #external-data {ANY_VALUE}},
   PERM   Layout-style                {STYLE_ID_OF(LStyleT8)}
}
```

7.3.3.30 Row

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$RowGFS},
   REQ    Application-comments        {REQ #constraint-name {"31"},
                                       PERM #external-data {ANY_VALUE}},
   REQ    Layout-style                {STYLE_ID_OF(LStyleT5)}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(Row)},
   REQ    Subordinates                {SUB_ID_OF(EntryElement)+},
```

```
   PERM   Application-comments        {REQ #constraint-name {"31"},
                                       PERM #external-data {ANY_VALUE}}
}
```

7.3.3.31 EntryElement

```
:COMP-LOGICAL {
GENERIC:
   REQ    Generator-for-subordinates  {$EntryElementGFS},
   REQ    Application-comments        {REQ  #constraint-name {"35"},
                                       PERM #external-data   {ANY_VALUE}},
   REQ    Layout-style                {STYLE_ID_OF(LStyleT2)}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(EntryElement)},
   REQ    Subordinates                {SUB_ID_OF(EntryText),
                                       SUB_ID_OF(EntryRaster),
                                       SUB_ID_OF(EntryGeometric)},
   PERM   Application-comments        {REQ  #constraint-name {"35"},
                                       PERM #external-data  {ANY_VALUE}}
}
```

7.3.3.32 EntryText

```
:BASIC-LOGICAL {
GENERIC:
   PERM   Resource                          {ANY_VALUE},
   REQ    Application-comments        {REQ  #constraint-name {"41"},
                                       PERM #external-data   {ANY_VALUE}}
SPECIFIC:
   REQ    Object-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. --
}
```

7.3.3.33 EntryRaster

```
:BASIC-LOGICAL {

GENERIC:
   REQ    Content-architecture-class  {$FPR},
   PERM   Resource                    {ANY_VALUE},
   REQ    Application-comments        {REQ  #constraint-name {"42"},
                                       PERM #external-data   {ANY_VALUE}}
SPECIFIC:
```

```
   REQ    Object-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. --
}
```

7.3.3.34 EntryGeometric

```
:BASIC-LOGICAL {
GENERIC:
   REQ    Content-architecture-class  {$FPG},
   PERM   Resource                    {ANY_VALUE},
   REQ    Application-comments        {REQ  #constraint-name {"43"},
                                       PERM #external-data   {ANY_VALUE}}
SPECIFIC:
   REQ    Object-class                {OBJECT_CLASS_ID_OF(EntryGeometric)},
   PERM   Content-architecture-class  {$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>     ::= <pageset-1>
                                        |<pageset-2>;
                                        |<pageset-3>;
                                        |SEQ(<pageset-1><pageset-2>);
                                        |SEQ(<pageset-1><pageset-3>);

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

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

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


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

 <headerarea>            ::= OBJECT_CLASS_ID_OF(BasicHeader)
```

```
                          | OBJECT_CLASS_ID_OF(CompositeHeader);

 <bodyarea>              ::= OBJECT_CLASS_ID_OF(VariableCompositeBody)
                          | OBJECT_CLASS_ID_OF(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(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>}...)
                          | CHO({<construction-term>}...);

 <construction-factor1>  ::= OBJECT_CLASS_ID_OF(BasicFloat)
                          | OBJECT_CLASS_ID_OF(SnakingColumns)
                          | OBJECT_CLASS_ID_OF(SynchronizedColumns)
                          | OBJECT_CLASS_ID_OF(TableArea)
                          | <construction-type>;

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


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

```
")


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

")


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)...);
")

DEFINE(TableHeaderGFS,"
 <construction-expr>      ::= SEQ(OBJECT_CLASS_ID_OF(SourcedContentFixed)...);
")


DEFINE(TableLabelGFS,"
 <construction-expr>      ::= SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)...);
")
```

7.4.2 Factor constraints

7.4.2.1 FACTOR ANY-LAYOUT

```
{
GENERIC:
   REQ    Object-type                  {VIRTUAL},
   REQ    Object-class-identifier      {ANY_VALUE}
SPECIFIC:
   PERM   Object-type                  {VIRTUAL},
   REQ    Object-identifier            {ANY_VALUE},
   CASE $DAC OF {
         $FDA:  PERM  Object-class     {VIRTUAL}
         $FPDA: REQ   Object-class     {VIRTUAL}
         },
   REQ    Subordinates                 {VIRTUAL}
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       {$USEPGNUMBERS}
         }
SPECIFIC:
   PERM   Object-type                  {'page'},
   REQ    Subordinates                 {SUB_ID_OF(BasicHeader),
                                       SUB_ID_OF(CompositeHeader),
                                       SUB_ID_OF(BasicBody),
                                       SUB_ID_OF(VariableCompositeBody),
                                       SUB_ID_OF(BasicFooter),
                                       SUB_ID_OF(CompositeFooter)}
SPECIFIC_AND_GENERIC:
   PERM   Dimensions                   {$PermissiblePageDimensions},
   PERM   Page-position                {ANY_VALUE}
}
```

7.4.2.3 FACTOR ANY-FRAME-FIXED

```
:ANY-LAYOUT {
GENERIC:
```

```
   REQ    Object-type                {'frame'}
SPECIFIC:
   PERM   Object-type                {'frame'}
SPECIFIC_AND_GENERIC:
   PERM   Position                   {REQ #fixed-position
                                       {REQ #horizontal-position     {ANY_VALUE},
                                        REQ #vertical-position       {ANY_VALUE}}},
   PERM   Dimensions                 {REQ #horizontal-dimension
                                      {REQ #fixed-dimension          {ANY_VALUE}},
                                     REQ #vertical-dimension
                                      {REQ #fixed-dimension          {ANY_VALUE}}},
   PERM   Border                     {ANY_VALUE}
}
```

7.4.2.4 FACTOR ANY-FRAME-VARIABLE

```
:ANY-LAYOUT {
GENERIC:
   REQ    Object-type                {'frame'}
SPECIFIC:
   PERM   Object-type                {'frame'},
   CASE $DAC OF {
        $FPDA:
                REQ Position         {REQ #fixed-position
                                        {REQ #horizontal-position      {ANY_VALUE},
                                        REQ #vertical-position         {ANY_VALUE}}},
                REQ Dimensions       {REQ #horizontal-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}},
                                     REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}}}
        }
SPECIFIC_AND_GENERIC:
   CASE $DAC OF {
        $FDA:
                PERM Position        {REQ #fixed-position
                                        {REQ #horizontal-position      {ANY_VALUE},
                                         REQ #vertical-position        {ANY_VALUE}}},
                PERM Dimensions      {REQ #horizontal-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}},
                                     REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}}}
        },
   PERM   Border                     {ANY_VALUE}
}
```

7.4.3 Constituent constraints

7.4.3.1 DocumentLayoutRoot

```
:ANY-LAYOUT {
GENERIC:
   REQ   Object-type                 {'document-layout-root'},
```

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

7.4.3.2 PageSet

```
:ANY-LAYOUT {
GENERIC:
   REQ    Object-type                  {'page-set'},

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

7.4.3.3 Page

```
:ANY-PAGE {
GENERIC:
   REQ    Application-comments        {REQ #constraint-name {"2"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
   CASE $DAC OF {
         $FDA:
               PERM Object-class      {OBJECT_CLASS_ID_OF(Page)}
         $FPDA:
               REQ  Object-class      {OBJECT_CLASS_ID_OF(Page)}
        },
   PERM   Application-comments        {REQ #constraint-name {"2"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
   PERM   Medium-type                 {PERM #nominal-page-size {$NominalPageSizes},
                                       PERM #side-of-sheet {ANY_VALUE}}
}
```

7.4.3.4 RectoPage

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

7.4.3.5 VersoPage

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

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

7.4.3.6 CompositeHeader

```
:ANY-FRAME-FIXED {
GENERIC:
   CASE $DAC OF {
        $PDA-FPDA:
                REQ    Generator-for-subordinates       {$HeaderFooterGFS}},
   PERM   Layout-path                   {'270-degrees'-- H/F layouts A1 --
                                         |'180-degrees' -- H/F layout B1 --},
   REQ    Application-comments          {REQ  #constraint-name {"5"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
   CASE $DAC OF {
        $FDA:
                PERM Object-class       {OBJECT_CLASS_ID_OF(CompositeHeader)}
        $FPDA:
                REQ  Object-class       {OBJECT_CLASS_ID_OF(CompositeHeader)}
        },
   REQ    Subordinates                  {SUB_ID_OF(SourcedContentFixed)+,
                                         SUB_ID_OF(ArrangedContentFixed)+,
                                         SUB_ID_OF(SourcedContentVariable)+,
                                         SUB_ID_OF(ArrangedContentVariable)+},
   PERM   Application-comments          {REQ  #constraint-name {"5"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.7 CompositeFooter

```
:ANY-FRAME-FIXED {
GENERIC:
   CASE $DAC OF {
        $PDA-FPDA:
                REQ    Generator-for-subordinates       {$HeaderFooterGFS}},
   PERM   Layout-path                   {'270-degrees' -- H/F layouts A1 --
                                         |'180-degrees' -- H/F layout B1 --},
   REQ    Application-comments          {REQ  #constraint-name {"32"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
   CASE $DAC OF {
        $FDA:
                PERM  Object-class      {OBJECT_CLASS_ID_OF(CompositeFooter)}
        $FPDA:
```

```
                REQ   Object-class      {OBJECT_CLASS_ID_OF(CompositeFooter)}
        },
    REQ   Subordinates                  {SUB_ID_OF(SourcedContentFixed)+,
                                         SUB_ID_OF(ArrangedContentFixed)+,
                                         SUB_ID_OF(SourcedContentVariable)+,
                                         SUB_ID_OF(ArrangedContentVariable)+},
    PERM   Application-comments         {REQ  #constraint-name {"32"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.8 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 --}
        },
    REQ   Application-comments          {REQ  #constraint-name {"7"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class      {OBJECT_CLASS_ID_OF
                                                 (VariableCompositeBody)}
        $FPDA:
                REQ   Object-class      {OBJECT_CLASS_ID_OF
                                                 (VariableCompositeBody)}
        },
    REQ   Subordinates                  {SUB_ID_OF(SnakingColumns)+,
                                         SUB_ID_OF(SynchronizedColumns)+,
                                         SUB_ID_OF(BasicFloat)+,
                                         SUB_ID_OF(FootnoteArea)+,
                                         SSUB_ID_OF(TableArea)+},
    PERM   Application-comments         {REQ  #constraint-name {"7"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.9 ColumnFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                PERM  Permitted-categories    {ANY_STRING...},
                REQ   Position          {REQ #fixed-position
                                              {REQ #horizontal-position {ANY_VALUE},
                                               REQ #vertical-position {ANY_VALUE}}},
                CASE SUPERIOR ({VariableCompositeBody} (Layout-path)) OF {
```

```
                {'270-degrees'}: -- 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 {"8"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class    {OBJECT_CLASS_ID_OF(ColumnFixed)}
        $FPDA:
                REQ   Object-class    {OBJECT_CLASS_ID_OF(ColumnFixed)}
        },
    REQ   Subordinates                {SUB_ID_OF(SpecificBlock)+},
    PERM  Application-comments        {REQ  #constraint-name {"8"},
                                       PERM #external-data {ANY_VALUE}}
}
```

7.4.3.10 ColumnVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                PERM  Permitted-categories    {ANY_STRING...},
                REQ   Position          {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
                CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
```

```
                {'270-degrees'}:  -- 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)}
        },
    REQ   Subordinates                {SUB_ID_OF(SpecificBlock)+},
    PERM  Application-comments        {REQ  #constraint-name {"9"},
                                       PERM #external-data {ANY_VALUE}}
}
```

7.4.3.11 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'}

        }  },
    REQ   Application-comments       {REQ  #constraint-name {"10"},
                                      PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class   {OBJECT_CLASS_ID_OF(Snakingcolumns)}
        $FPDA:
                REQ   Object-class   {OBJECT_CLASS_ID_OF(Snakingcolumns)}
        },
    REQ   Subordinates               {SUB_ID_OF(ColumnVariable)+},
    PERM  Application-comments       {REQ  #constraint-name {"10"},
                                      PERM #external-data {ANY_VALUE}}
}
```

7.4.3.12 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'}
        }   },
    REQ   Application-comments        {REQ  #constraint-name {"11"},
                                       PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM Object-class     {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
        $FPDA:
                REQ  Object-class     {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
        },
    REQ   Subordinates                {SUB_ID_OF(ColumnFixed)+},
    PERM  Application-comments        {REQ #constraint-name {"11"},
                                       PERM #external-data {ANY_VALUE}}
}
```

7.4.3.13 BasicFloat

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                REQ   Position          {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'normal-order'}}},
                PERM  Permitted-categories    {ANY_STRING...},

                CASE SUPERIOR ({VariableCompositeBody} (Layout-path)) OF {
```

```
                {'270-degrees'}: -- 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'}
        }   },
    REQ    Application-comments         {REQ  #constraint-name {"12"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class   {OBJECT_CLASS_ID_OF(BasicFloat)}
        $FPDA:
                REQ   Object-class   {OBJECT_CLASS_ID_OF(BasicFloat)}
        },
    REQ    Subordinates                 {SUB_ID_OF(SpecificBlock)+},
    PERM   Application-comments         {REQ  #constraint-name {"12"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.14 FootnoteArea

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                REQ   Position          {REQ #variable-position {
                                                PERM #offset     {ANY_VALUE},
                                                PERM #separation {ANY_VALUE},
                                                PERM #alignment  {ANY_VALUE},
                                                PERM #fill-order {'reverse-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'}
        }   },
    REQ   Permitted-categories  {$FOOTNOTECATEGORY},
    REQ   Application-comments          {REQ  #constraint-name {"15"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class      {OBJECT_CLASS_ID_OF(FootnoteArea)}
        $FPDA:
                REQ   Object-class      {OBJECT_CLASS_ID_OF(FootnoteArea)}
        },
    REQ    Subordinates                 {SUB_ID_OF(SpecificBlock)+},
    PERM   Application-comments         {REQ  #constraint-name {"15"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.15 ArrangedContentFixed

```
:ANY-FRAME-FIXED {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                REQ  Generator-for-subordinates
                                {<construction-expr>::=SEQ(OBJECT_CLASS_ID_OF(GenericBlock)...);}},
    REQ    Application-comments         {REQ  #constraint-name {"16"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class      {OBJECT_CLASS_ID_OF
                                                (ArrangedContentFixed)}
```

```
        $FPDA:
                REQ   Object-class      {OBJECT_CLASS_ID_OF
                                                (ArrangedContentFixed)}
        },
    REQ    Subordinates                 {SUB_ID_OF(GenericBlock)+},
    PERM   Application-comments         {REQ  #constraint-name {"16"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.16 ArrangedContentVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                REQ    Generator-for-subordinates {<construction-expr>::=
                                        SEQ(OBJECT_CLASS_ID_OF(GenericBlock)...)},
                REQ   Position          {REQ #variable-position {
                                        PERM #offset     {ANY_VALUE},
                                        PERM #separation {ANY_VALUE},
                                        PERM #alignment  {ANY_VALUE},
                                        PERM #fill-order {'normal-order'}}},
                REQ   Dimensions        {REQ #horizontal-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}},
                                         REQ #vertical-dimension
                                        {REQ #fixed-dimension {ANY_VALUE}}}
        },
    REQ    Application-comments         {REQ  #constraint-name {"17"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
                PERM  Object-class      {OBJECT_CLASS_ID_OF
                                                (ArrangedContentVariable)}
        $FPDA:
                REQ   Object-class      {OBJECT_CLASS_ID_OF
                                                (ArrangedContentVariable)}
        },
    REQ    Subordinates                 {SUB_ID_OF(GenericBlock)+},
    PERM   Application-comments         {REQ  #constraint-name {"17"},
                                         PERM #external-data {ANY_VALUE}}
}
```

7.4.3.17 SourcedContentFixed

```
:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
                REQ    Logical-source           {OBJECT_CLASS_ID_OF(CommonContent)},
                REQ    Position                 {REQ #fixed-position
                                                  {REQ #horizontal-position {ANY_VALUE},
```

```
                                        REQ #vertical-position {ANY_VALUE}}},
            REQ   Dimensions            {REQ #horizontal-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}},
                                        REQ #vertical-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #rule-b {ANY_VALUE}}},

            CASE SUPERIOR ({CompositeHeader | CompositeFooter |
                            TableHeader} (Layout-path)) OF {

            {'270-degrees'}:    -- H/F layout A1 when the immediate superior is
                                CompositeHeader, CompositeFooter or TableHeader --
                PERM Layout-path    {'270-degrees'}  -- H/F layout A1 --

            {'180-degrees'}:    -- H/F layout B1 when the immediate superior is
                                CompositeHeader or CompositeFooter --
                REQ  Layout-path    {'180-degrees'} -- H/F layout B1 --
        }   },
   REQ   Application-comments   {REQ  #constraint-name {"18"},
                                PERM #external-data {ANY_VALUE}}
SPECIFIC:
   CASE $DAC OF {
        $FDA:
            PERM  Object-class  {OBJECT_CLASS_ID_OF
                                        (SourcedContentFixed)}
        $FPDA:
            REQ   Object-class  {OBJECT_CLASS_ID_OF
                                        (SourcedContentFixed)}
        },
   REQ   Subordinates           {SUB_ID_OF(SpecificBlock)+},
   PERM  Application-comments   {REQ  #constraint-name {"18"},
                                 PERM #external-data {ANY_VALUE}}
}
```

7.4.3.18 SourcedContentVariable

```
:ANY-FRAME-VARIABLE {
GENERIC:
   CASE $DAC OF {
        $PDA-FPDA:
            REQ  Logical-source  {OBJECT_CLASS_ID_OF(CommonContent)},
            REQ  Position        {REQ #variable-position {
                                        PERM #offset     {ANY_VALUE},
                                        PERM #separation {ANY_VALUE},
                                        PERM #alignment  {ANY_VALUE},
                                        PERM #fill-order {'normal-order'}}},

            CASE SUPERIOR ({CompositeHeader | CompositeFooter} (Layout-path)) OF {

            {'270-degrees'}:    -- H/F layout A1 when the immediate superior is
                                CompositeHeader or CompositeFooter, or --
                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'}  -- H/F layout A1 --

            {'180-degrees'}:        -- H/F layout B1 when the immediate superior is
                                    CompositeHeader or CompositeFooter, or --
                    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'}  -- H/F layout B1 --
        }   },

   REQ   Application-comments       {REQ  #constraint-name {"19"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
   CASE $DAC OF {
        $FDA:
              PERM  Object-class    {OBJECT_CLASS_ID_OF
                                          (SourcedContentVariable)}
        $FPDA:
              REQ   Object-class    {OBJECT_CLASS_ID_OF
                                          (SourcedContentVariable)}
        },
   REQ   Subordinates               {SUB_ID_OF(SpecificBlock)+},
   PERM  Application-comments       {REQ  #constraint-name {"19"},
                                     PERM #external-data {ANY_VALUE}}
}
```

7.4.3.19 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}}
}
```

7.4.3.20 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}}
}
```

7.4.3.21 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}}
}
```

7.4.3.22 GenericBlock

```
{
GENERIC:
   REQ    Object-type                  {'block'},
   REQ    Object-class-identifier      {ANY_VALUE},
   REQ    Content-architecture-class   {$FC | $FPC | $FPR | $FPG },
   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 | $FPG },
   CASE  GenericBlock (Object-class) OF {
        VOID:
                REQ   Content-portions    {CONTENT_ID_OF(Character-content-portion)+
                                          |CONTENT_ID_OF(Raster-graphics-content-portion)
                                          |CONTENT_ID_OF(Geometric-graphics-content-portion)}
        },
   PERM  Presentation-attributes       {
      PERM  #character-attributes      {
         PERM  #alignment                      {ANY_VALUE},
         PERM  #character-fonts                {ANY_VALUE},
         PERM  #character-path                 {ANY_VALUE},
         PERM  #character-spacing              {ANY_VALUE},
         PERM  #character-orientation          {'0-degrees'|'90-degrees'},
         PERM  #code-extension-announcers      {$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               {'90-degrees'|'270-degrees'},
         PERM  #line-spacing                   {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  User-readable-comments       {ANY_STRING},
   PERM  User-visible-name            {ANY_STRING}
}
```

7.4.3.23 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 | $FPG},
   PERM  Presentation-attributes      {
      PERM  #character-attributes     {
         PERM  #alignment                        {ANY_VALUE},
         PERM  #character-fonts                  {ANY_VALUE},
         PERM  #character-path                   {ANY_VALUE},
         PERM  #character-spacing                {ANY_VALUE},
         PERM  #character-orientation            {'0-degrees'|'90-degrees'},
         PERM  #code-extension-announcers        {$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                 {'90-degrees'|'270-degrees'},
         PERM  #line-spacing                     {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}}
}
```

7.4.3.24 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  Border                    {ANY_VALUE}
}
```

7.4.3.25 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  Border                         {ANY_VALUE}
}
```

7.4.3.26 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)},
   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  Border                      {ANY_VALUE}
}
```

7.4.3.27 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)+,
    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  Border                        {ANY_VALUE}
}
```

7.4.3.28 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  Application-comments          {REQ  #constraint-name {"41"},
                                         PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM  Border                        {ANY_VALUE}
}
```

7.4.3.29 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}}
}
```

7.5 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 --
")
```

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 constraint Passage  --

CASE $GLAS OF{
  $COMPLETE:
     PERM       Indivisibility                  {ANY_VALUE},
     PERM       Layout-object-class             {OBJECT_CLASS_ID_OF(PageSet)},
     PERM       New-layout-object               {OBJECT_CLASS_ID_OF(PageSet)},
  VOID:
     PERM       Indivisibility                  {ANY_STRING --to layout category--
                                                |'page'     --to layout object type--
                                                |'null'},
}}
```

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    New-layout-object        {ANY_VALUE},
    PERM    Offset                   {ANY_VALUE},
    PERM    Same-layout-object       {$SameLayoutObject},
    PERM    Separation               {PERM #leading-edge {ANY_INTEGER},
                                            PERM #trailing-edge {ANY_INTEGER}},
    PERM    Synchronization          {ANY_VALUE}
  VOID:
    PERM    Block-alignment          {ANY_VALUE},
    PERM    Concatenation            {ANY_VALUE},
    PERM    Indivisibility           {ANY_STRING --to layout category--
                                     |'page'     --to layout object type--
                                     |'null'},
    PERM    Layout-category          {ANY_VALUE},
    PERM    New-layout-object        {ANY_STRING --to layout category--
                                     |'page'     --to layout object type--
                                     |'null'},
    PERM    Offset                   {ANY_VALUE},
    PERM    Same-layout-object       {$SameLayoutObject},
    PERM    Separation               {PERM #leading-edge {ANY_INTEGER},
                                            PERM #trailing-edge {ANY_INTEGER}},
    PERM    Synchronization          {ANY_VALUE}
 }}
```

7.5.3.3 LStyle3

```
:ANY-LAYOUT-STYLE{

-- this style is used for the constituent constraints CommonText, PageNumber, TableNumber and
CommonReference  --

PERM  Block-alignment          {ANY_VALUE},
PERM  Concatenation            {ANY_VALUE},
PERM  Offset                   {ANY_VALUE},
PERM  Separation               {PERM #leading-edge {ANY_INTEGER},
                                      PERM #trailing-edge {ANY_INTEGER}}
}
```

7.5.3.4 LStyle4

```
:ANY-LAYOUT-STYLE  {

-- this style is used for the constituent constraints NumberedSegment, Paragraph, Phrase, Reference,
NumberedList, UnNumberedList, DefinitionList, ListItem, ListTerm  --

CASE $GLAS OF{
  $COMPLETE:
    PERM    Indivisibility           {ANY_VALUE},
    PERM    Layout-object-class      {OBJECT_CLASS_ID_OF(PageSet)},
    PERM    New-layout-object        {ANY_VALUE},
```

```
    PERM      Same-layout-object          {$SameLayoutObject},
    PERM      Synchronization             {ANY_VALUE}
  VOID:
    PERM      Indivisibility              {ANY_STRING --to layout category--
                                          |'page'     --to layout object type--
                                          |'null'},
    PERM      New-layout-object           {ANY_STRING --to layout category--
                                          |'page'     --to layout object type--
                                          |'null'},
    PERM      Same-layout-object          {$SameLayoutObject},
    PERM      Synchronization             {ANY_VALUE}
}}
```

7.5.3.5 LStyle5

```
:ANY-LAYOUT-STYLE  {

-- This style is used for the constituent constraints BodyRaster and BodyGeometric  --

CASE $GLAS OF{
  $COMPLETE:
    PERM      Block-alignment             {ANY_VALUE},
    PERM      Layout-category             {ANY_VALUE},
    PERM      New-layout-object           {ANY_VALUE},
    PERM      Offset                      {ANY_VALUE},
    PERM      Same-layout-object          {$SameLayoutObject},
    PERM      Separation                  {PERM #leading-edge {ANY_INTEGER},
                                                PERM #trailing-edge {ANY_INTEGER}},
    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          {$SameLayoutObject},
    PERM      Separation                  {PERM #leading-edge {ANY_INTEGER},
                                                PERM #trailing-edge {ANY_INTEGER}},
    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              {OBJECT_CLASS_ID_OF(FootnoteArea)
```

```
                                                |'page'
                                                |'null'},
    PERM    Block-alignment             {ANY_VALUE},
    PERM    Concatenation               {ANY_VALUE},
    REQ     Layout-category             {$FOOTNOTECATEGORY},
    PERM    Offset                      {ANY_VALUE},
    PERM    Separation                  {PERM #leading-edge {ANY_INTEGER},
                                            PERM #trailing-edge {ANY_INTEGER}}
  VOID:
    PERM    Indivisibility              {ANY_STRING --to layout category--
                                        |'page'     --to layout object type--
                                        |'null'},
    PERM    Block-alignment             {ANY_VALUE},
    PERM    Concatenation               {ANY_VALUE},
    REQ     Layout-category             {$FOOTNOTECATEGORY},
    PERM    Offset                      {ANY_VALUE},
    PERM    Separation                  {PERM #leading-edge {ANY_INTEGER},
                                            PERM #trailing-edge {ANY_INTEGER}}
}}
```

7.5.3.7 LStyle7

```
:ANY-LAYOUT-STYLE  {

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

PERM  Same-layout-object          {$SameLayoutObject},
}
```

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                  {PERM #leading-edge {ANY_INTEGER},
                                        PERM #trailing-edge {ANY_INTEGER}}
}
```

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                    {PERM #leading-edge {ANY_INTEGER},
                                                PERM #trailing-edge {ANY_INTEGER}}
}
```

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               {PERM #leading-edge {ANY_INTEGER},
                                                PERM #trailing-edge {ANY_INTEGER}}
  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       {$SameLayoutObject},
     PERM      Separation               {PERM #leading-edge {ANY_INTEGER},
                                                PERM #trailing-edge {ANY_INTEGER}}
}}
```

7.5.3.11 LStyle11

```
:ANY-LAYOUT-STYLE  {

-- this style is used for the constituent constraint FootnoteBody  --

CASE $GLAS OF{
  $COMPLETE:
     PERM      Indivisibility           {OBJECT_CLASS_ID_OF(FootnoteArea)
                                                |'page'
                                                |'null'},
     PERM      Same-layout-object       {$SameLayoutObject},
  VOID:
     PERM      Indivisibility           {ANY_STRING --to layout category--
                                        |'page'     --to layout object type--
                                        |'null'},
     PERM      Same-layout-object       {$SameLayoutObject},
}}
```

7.5.3.12 LStyleT2

```
:ANY-LAYOUT-STYLE  {

-- this style is used for the constituent constraint EntryElement  --

-- 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 --  }
}
```

7.5.3.13 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}
}
```

7.5.3.14 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'}
}
```

7.5.3.15 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}
}
```

7.5.3.16 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},
}
```

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              {'0-degrees'|'90-degrees'},
        PERM  #character-path                     {ANY_VALUE},
        PERM  #character-spacing                  {ANY_VALUE},
        PERM  #code-extension-announcers          {$CDEXTEN},
        PERM  #first-line-offset                  {ANY_VALUE},
        PERM  #graphic-character-sets             {$PERMIT-GRCHAR},
        PERM  #graphic-character-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  #proportional-line-spacing          {ANY_VALUE},
        PERM  #widow-size                         {ANY_VALUE},
    }
}}
```

7.6.3.2 PStyle2

```
:ANY-PRESENTATION-STYLE {

--      NOTE: This style is named PStyle4 in FOD26 | PM26. --

--      This style is used for the constituent constraints BodyGeometric, CommonGeometric,
        EntryGeometric, GenericBlock and SpecificBlock. --

PERM  Presentation-attributes  {
    PERM #geometric-graphics-attributes {
        PERM  #picture-dimensions           {ANY_VALUE},
        PERM  #picture-orientation      {ANY_VALUE},
        PERM  #text-rendition           {PERM #fonts-list {ANY_VALUE},
                                                PERM #character-set-list {ANY_VALUE}}
    }
}}
```

7.6.3.3 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-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 {

--      NOTE: This style is named PStyle2 in FOD26 | PM26. --

--      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            {'0-degrees'|'90-degrees'},
        PERM  #character-path                   {'0-degrees'|'180-degrees'|'270-degrees'},
        PERM  #character-spacing                {ANY_VALUE},
        PERM  #code-extension-announcers        {$CDEXTEN},
        PERM  #first-line-offset                {ANY_VALUE},
        PERM  #graphic-character-sets           {$PERMIT-GRCHAR},
        PERM  #graphic-character-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  #proportional-line-spacing        {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
         | #LF
         | #PLD
         | #PLU
         | #SCS                         {ANY_VALUE}
         | #SGR                         {$GRAPHICRENDITIONS}
         | #SHS                         {0,1,2,3,4}
         | #SLS                         {ANY_VALUE}
         | #SRS                         {ANY_VALUE}
         | #STAB                        {ANY_VALUE}
         | #SUB
         | #SVS                         {ANY_VALUE}
```

```
        | #VPB                          {ANY_VALUE}
        | #VPR                          {ANY_VALUE}

-- Layout Control Functions --
        | #JFY                          {0}

-- Logical Control Functions --
        | #BPH
        | #NBH

-- 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 Specification, the interchange format class A shall be used when applying Office Document Interchange Format (ODIF), and the interchange format SGML Document Interchange Format (SDIF) shall be used when applying Office Document Language (ODL) in conjunction with SDIF.

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 }

This is the object identifier used to represent the FOD36 | PM-36 documentation application profile. The re is no document profile attribute intended for the purpose of identifying documents which conform to an Implementation Support Requirements on a DAP, as in the case of this Specification. However, applicat ions must have some means of distinguishing fully-functional FOD36 | PM-36 documents from FOD36 | PM-36 documents which adhere to the Implementation Support Requirements specified in this document. To do this, the document profile document management attribute "Additional-information" is required by this specification to have the value "Level 2 to Level 3 Migration DAP".

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 encodi ng 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 | ISO/IEC 8824, in data streams which may be encoded in accordance with this document application pr ofile is 32 767 octets. If it is required to encode an octet string of greater length than this, constructed t ype 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.

8.2 Interchange format SDIF

NOTE - The subclause 8.2 applies to the International Standardized Profile only.

8.2.1 Interchange format

The document profile attribute "interchange format" does not apply for this interchange format. This form of the interchange format is defined in Annex E of CCITT Rec. T.415 | ISO 8613-5. CCITT Rec. T.416 | ISO 8613-6, CCITT Rec. T.417 | ISO 8613-7 and CCITT Rec. T.418 | ISO 8613-8 also contain additional specifications for this interchange format.

8.2.2 Document application profile object identifier

The value for the attribute "document application profile" for this interchange format is represented by the following object identifier:

ASN.1 { 1 0 11182 0 36 0 }

NOTE - There is no requirement to include a part number arc within the object identifier for the document application profile.

This is the object identifier used to represent the FOD36 | PM-36 documentation application profile. The re is no document profile attribute intended for the purpose of identifying documents which conform to an Implementation Support Requirements on a DAP, as in the case of this Specification. However, applicat ions must have some means of distinguishing fully-functional FOD36 | PM-36 documents from FOD36 | PM-36 documents which adhere to the Implementation Support Requirements specified in this document. To do this, the document profile document management attribute "Additional-information" is required by this specification to have the value "Level 2 to Level 3 Migration DAP".

8.2.3 Encoding of application comments

The encoding of the attribute "application comments" is defined in this encoding as a data stream conforming to this Specification with the following document type definition (DTD):

```
<!-- Public document type definition.  Typical invocation:
     <!DOCTYPE fodapc PUBLIC "ISO/IEC 11182-1 : 1993//DTD
                              Application Comments//EN"  [   ]>
     -->
     <!ELEMENT fodapc     - 0      (externl?)>
     <!ATTLIST fodapc     consname CDATA   #IMPLIED>
     <!ELEMENT externl    - 0      (#PCDATA)>
     <!ATTLIST externl    loc      ENTITY          #CONREF>
```

For example, a typical SUBDOC for representing the "application comments" of a Paragraph then would look as follows:

```
<!DOCTYPE fodapc PUBLIC "ISO/IEC 11182-1 : 1993//DTD Application Comments//EN">
<fodapc consname="6">
```

If the optionality of the attribute "fodapc" is specified in an earlier portion of the DTD, the invocation may be minimised further because the tag is not needed when application comment is not included as is the case here.

The content of external data appearing inline is restricted to parsable data. Referenced external data n eed not be parsable.

Annex A

Amendments and technical corrigenda

(This annex forms an integral part of this Specification.)

A.1 Amendments

A.1.1 Amendments to the base standard

The amendments applicable to this Specification include text to be included in CCITT Rec. T.411 | ISO 8613-1 as the following annexes:

— Annex E: Use of ISO/IEC 10021 (MOTIS) to interchange documents conforming to CCI TT Rec. T.410 series | ISO 8613 - published as [First extension to the CCITT Rec. T.41 0 series (1988), Addendum II | ISO 8613-1 (1989): Amendment 1];

— Annex F: Document Application Profile proforma and notation - published as CCITT Re c. T.411 - Annex F (1991) | ISO 8613-1 (1989): Addendum 1;

— Annex G: Conformance testing methodology - published as ISO 8613-1 (1989): Amendment 2;

— Annex H: Recording of documents conforming to CCITT Rec. T.410 series | ISO 8613 o n flexible disk cartridges conforming to ISO 9293 - currently Draft Amendment 5 to ISO 8613-1 (1989) - Annex H.

This Specification does not include the following features of the amendment:

— Addendum on security - published as [First extension to the CCITT Rec. T.410 series (1988), Addendum IV | ISO 8613 (1989) Addendum 4];

— Addendum on styles - published as [First extension to the CCITT Rec. T.410 series (19 88), Addendum IV | ISO 8613 (1989) Addendum 6];

— Addendum on alternative representation - published as [First extension to the CCITT R ec. T.410 series (1988), Addendum III | ISO 8613 (1989) Addendum 3];

— Addendum on colour - published as [Revision of the CCITT Rec. T.410 series containe d in COM VIII - R30-E Addendum 2 | ISO 8613 (1989) - Amendment 2];

— Addendum on tiled raster graphics - published as [First extension to the CCITT Rec. T. 410 series (1988) Addendum I | ISO 8613 (1989) - Addendum 1];

— Addendum on streams - published as ISO 8613 (1989) - Amendment 5.

In addition, this amendment addresses the inclusion of the CCITT Rec. 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 Specification

There is no technical corrigendum specific to this Specification.

A.3 Versions of the ODA Specification

The version of the ODA Specification defined in clause A.1 is known as "version 1.1". The associated date is 1992-01-01. The version of the ODA Specification including all features of the amendments, addenda and technical corrigenda mentioned in clause 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 Rec. T.410 series (1988) | ISO 8613 (1989); version 1.1"
— publication date: "1992-01-01"

Annex B

Recommended practices

(This annex does not form an integral part of this Specification.)

B.1 Transfer methods for ODA

B.1.1 Conveyance of ODA over CCITT X.400-1984

This describes how ODA body parts are to be encoded for transmission over a CCITT X.400-1984 ser vice.

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 Specification.

B.1.2 Conveyance of ODA over FTAM

This describes the FTAM Document Type to be used for minimal storage and transfer capabilities of O DA 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 ty pe 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 cont

ains an ODA data stream.

B.1.3 Conveyance of ODA over DTAM

This provides for information concerning the interchange of ODA based documents with DTAM protocol s.

DTAM (Document Transfer and Manipulation) is defined in the T.430 series of CCITT Recommendations and is, like ODA, an integral part of the T.400 series of CCITT Recommendations named Open Docum ent Architecture, Transfer and Manipulation.

The T.520 series of CCITT Recommendations contain Communication Application Profiles (CAP). CCI TT Recommendation T.522 describes the Communication Application Profile BT1 for document bulk transfer. CCITT Recommendation T.522 is applicable for the Open Document Format Profile (FOD) published in this Specification.

NOTE - The use of BT1 within the end-to-end oriented Telematic Services Telefax 4 and Teletex is described in CCITT Recommendation T.561, subclause 7.1 and CCITT Recommendation T.562, subclause 7.1.

B.1.4 Conveyance of ODA over flexible disks

The recommended method for interchanging ODA documents between systems by the exchange of magnetically recorded Flexible Disk Cartridges is by the use of an annex to CCITT Rec. T.411 | ISO 86 13-1 (to be published), *Recording of Documents Conforming to ISO 8613 on Flexible Disk Cartridges Conforming to ISO 9293"*. This annex provides for recording each ODA document as a separate file as defined by ISO 9293.

NOTE - The following information applies to the International Standardized Profile only. *Documents encoded in Office Document Language (ODL) may be stored such that each SGML ENTITY is recorded in a separate file or in the case of SGML Document Interchange Format (SDIF) encoding, the file may be stored in a single file.*

B.2 Font reference

The recommended method for specifying a font reference is to base it on ISO/IEC 9541.

Font sizes from 6 to 72 points (100 to 1200 BMU) are intended to be supported by implementation conforming to this 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/IEC 9541 that are to be specified in a Font-Attribute-Set are those specified by the following document application profile notation.

```
Font-Attribute-Set      {

PERM  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/IEC 8632 (CGM) constraints for this document application profile

It is a recommended that geometric graphics content information contain only those elements listed in t his 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/IEC 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. T he sequence of zero or more octets is for padding purposes.
Begin Metafile	Support will be provided for strings with a length up to 254 octe ts, except for data records which will support strings with a leng th up to 32 767 octets.
End Metafile	
Begin Picture	Support will be provided for strings with a length up to 254 octe ts, except for data records which will support strings with a leng th 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 octe ts, except for data records which will support strings with a leng th up to 32 767 octets. The METAFILE DESCRIPTION string parameter will be used to include the sub-string "CCITT Rec. PM-36" or "ISO FOD36" to label the content information as conforming to this Specification. In addition, generators of cont ent 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/IEC 9541 names is preferred, but font names may be specified using proprietary fo nt names.
Character Set List	All character sets referenced in the metafile shall be defined in CHARACTER SET LIST. Permissible character sets are the same as for character content architecture.
Character Coding Announcer	--

B.3.3 Picture descriptor elements

Scaling Mode	The Scale Factor parameter of SCALING MODE element is always a 32-bit floating point value, even when the REAL PRECISION has selected fixed point for other real numbers. It is not apparent in ISO/IEC 8632 what the precision of this floating point value is when fixed point has been selected. Its precision shall be (0, 9, 23).
Colour Selection Mode	--
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 1 024 vertic es.
Disjoint Polyline	The minimum support for the length of point lists is 1 024 vertic es.
Polymarker	The minimum support for the length of point lists is 1 024 vertic es.
Text	Support will be provided for strings with a length up to 254 octe ts, except for data records which will support strings with a leng th 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 octe ts, except for data records which will support strings with a leng th 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 octe ts, except for data records which will support strings with a leng th 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 1 024 vertic es.
Polygon Set	The minimum support for the length of point lists is 1 024 vertic es.
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 1 0 24 x 1 024 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 Rec. T.410 series | ISO 8613 or in ISO/IEC 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 S ET LIST. The only character sets which may be designated in G0 are ISO/IEC 646 IRV or versions of ISO/IEC 646. Other charac ter 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 f or the PATTERN TABLE element is 2 048. This will support 8 patterns of 16 x 16, 2 patterns of 32 x 32 or 1 pattern of 32 x 6 4. 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 i t appears in a picture subsequent to any graphical primitives. T he 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 t he COLOUR TABLE element is 63. This will support a 64 (0..6 3) 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 octe ts, except for data records which will support strings with a leng th up to 32 767 octets.

B.4 Interoperability with SGML applications

NOTE - Annex B.4 applies to the International Standardized Profile only.

The recommended method for the exchange of documents between Standard Generalized Markup Language (SGML), ISO 8879, based systems and systems based on this ODA document application pr ofile is by means of exchanging a document representation conforming to these agreements in an enco ded form of the SGML language known as the Office Document Language (ODL). ODL is a standardized SGML application for representing documents conforming to the ODA base standard. Such a representation may be converted into the Office Document Interchange Format (ODIF) supported by this document application profile.

Annex C

Bibliography

(This annex does not form an integral part of this Specification.)

— CCITT Recommendation T.50 (1992), *International Reference Alphabet*.

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

— CCITT Recommendation T.52 [4], *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 (MOTIS)*.

[4]Presently at the stage of draft.