



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

T.506

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(08/93)

TERMINAUX POUR LES SERVICES TÉLÉMATIQUES

**PROFIL D'APPLICATION DE DOCUMENT
PM-36 POUR L'ÉCHANGE DE STRUCTURES
DOCUMENTAIRES ÉTENDUES ET DE
DOCUMENTS À CONTENU MIXTE SOUS
FORMES RETRAITABLE ET FORMATÉE**

Recommandation UIT-T T.506

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation UIT-T T.506, que l'on doit à la Commission d'études 8 (1993-1996) de l'UIT-T, a été approuvée le 31 août 1993 selon la procédure définie dans la Résolution n° 1 de la CMNT.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1995

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

		<i>Page</i>
1	Champ d'application.....	1
2	Références normatives	1
	2.1 Recommandations normes internationales couplées équivalentes par le contenu technique	1
	2.2 Références additionnelles	2
3	Définitions.....	3
4	Relations avec d'autres profils.....	4
5	Conformité	4
	5.1 Conformité de train de données	4
	5.2 Conformité d'application.....	4
6	Caractéristiques admises par le profil d'application documentaire.....	4
	6.1 Vue d'ensemble.....	5
	6.1.1 Considérations générales.....	5
	6.1.2 Documents sous forme formatée.....	5
	6.1.3 Documents sous forme retraitable.....	5
	6.1.4 Documents sous forme retraitable formatée.....	5
	6.1.5 Documents génériques	6
	6.1.6 Profil de document	6
	6.2 Caractéristiques logiques	6
	6.2.1 Introduction.....	6
	6.2.2 Vue d'ensemble de la structure logique.....	6
	6.2.3 Corps de la structure logique	7
	6.2.3.1 DocumentLogicalRoot (Racine logique de document)	7
	6.2.3.2 Passage (Passage).....	7
	6.2.3.3 NumberedSegment (Segment numéroté)	8
	6.2.3.4 Number (Numéro).....	8
	6.2.3.5 Paragraph (Paragraphe).....	9
	6.2.3.6 BodyText, BodyRaster et BodyGeometric (corps de texte, contenu graphique en points de corps de texte et contenu graphique géométrique de corps de texte).....	9
	6.2.3.7 Footnote (Note de bas de page)	10
	6.2.3.8 FootnoteReference (Renvoi à une note de bas de page)	10
	6.2.3.9 FootnoteBody (Corps de texte de note de bas de page)	10
	6.2.3.10 FootnoteNumber (Numéro de note de bas de page).....	11
	6.2.3.11 FootnoteText (Texte de note de bas de page)	11
	6.2.3.12 Title (Titre).....	11
	6.2.3.13 Phrase (Phrase).....	12
	6.2.3.14 Constituants prévoyant un mécanisme général de référence.....	12
	6.2.3.15 Constituants représentant des illustrations	13
	6.2.3.16 Constituants utilisés pour représenter des tableaux	15
	6.2.3.17 Constituants représentant des listes.....	16
	6.2.4 Contenu commun de la structure logique.....	18
	6.2.4.1 CommonContent (Contenu commun)	18
	6.2.4.2 CommonText (Texte commun).....	18
	6.2.4.3 PageNumber (Numéro de page).....	18
	6.2.4.4 CommonRaster (Contenu graphique en points commun)	18
	6.2.4.5 CommonGeometric (Contenu graphique géométrique commun)	18
	6.2.4.6 CommonReference (Référence commune).....	19
	6.2.4.7 CurrentInstance (Application courante).....	19
	6.2.4.8 TableNumber (Numéro de tableau)	19
	6.2.4.9 CommonNumber (Numéro commun)	19

	<i>Page</i>	
6.3	Caractéristiques physiques.....	19
6.3.1	Vue d'ensemble des caractéristiques physiques.....	19
6.3.2	Document LayoutRoot.....	20
6.3.3	PageSet (Ensemble de pages).....	20
6.3.4	Caractéristiques de page.....	20
6.3.4.1	Constituants de page.....	20
6.3.4.2	Dimensions des pages.....	21
6.3.4.3	Formats de page nominaux.....	21
6.3.4.4	Décalage de page.....	21
6.3.4.5	Caractéristiques physiques des pages.....	21
6.3.5	Caractéristiques de la zone de corps.....	25
6.3.5.1	Caractéristiques générales.....	25
6.3.5.2	BasicBody (Corps de base).....	25
6.3.5.3	VariableCompositeBody (Corps composite variable).....	26
6.3.5.4	BasicFloat (Colonne unique).....	26
6.3.5.5	SnakingColumns (Colonnes serpentines).....	29
6.3.5.6	SynchronizedColumns (Colonnes synchronisées).....	31
6.3.5.7	ColumnVariable (Colonne variable).....	32
6.3.5.8	ColumnFixed (Colonne fixe).....	32
6.3.5.9	FootnoteArea (Zone note de bas de page).....	33
6.3.5.10	FixedCompositeBody (Corps composite fixe).....	33
6.3.5.11	CompositeFloat (Élément mobile composite).....	34
6.3.5.12	CompositeFixtureVariable (Élément composite variable).....	35
6.3.5.13	CompositeFixtureFixed (Élément composite fixe).....	36
6.3.5.14	CompositeArtwork (Dessin composite).....	36
6.3.5.15	BasicFixture (Élément de base).....	36
6.3.5.16	FormArea (Zone formulaire).....	36
6.3.5.17	EntryGroupArea (Zone de groupe d'entrée).....	37
6.3.5.18	FormEntryArea (Zone entrée formulaire).....	37
6.3.5.19	BasicColumn (Colonne de base).....	38
6.3.5.20	CompositeColumnVariable (Colonne composite variable).....	38
6.3.5.21	CompositeColumnFixed (Colonne composite fixe).....	38
6.3.5.22	CompositeCommon (Composite commun).....	38
6.3.5.23	Constituants utilisés pour la présentation des tableaux.....	39
6.3.6	Caractéristiques des zones d'en-tête et de bas de page.....	44
6.3.6.1	Caractéristiques générales.....	44
6.3.6.2	BasicHeader et BasicFooter (En-tête de base et bas de page de base).....	44
6.3.6.3	CompositeHeader et CompositeFooter (En-tête composite et bas de page composite).....	44
6.3.6.4	SourcedContentVariable (Contenu variable provenant d'une source).....	45
6.3.6.5	ArrangedContentVariable (Contenu arrangé variable).....	46
6.3.6.6	SourcedContentFixed (Contenu fixe provenant d'une source).....	46
6.3.6.7	ArrangedContentFixed (Contenu arrangé fixe).....	46
6.3.7	GenericBlock et SpecificBlock (Pavé générique et pavé spécifique).....	46
6.4	Caractéristiques physiques du document.....	47
6.4.1	Contrôle de flux.....	47
6.4.1.1	Attribution du contenu aux ensembles de pages.....	47
6.4.1.2	Coupures de page.....	47
6.4.1.3	Attribution d'un contenu aux zones de corps.....	48
6.4.1.4	Attribution d'un contenu aux zones d'en-tête ou de bas de page.....	52
6.4.2	Mise en page du contenu du document.....	53
6.4.2.1	Marges.....	53
6.4.2.2	Séparation.....	54
6.4.2.3	Indivisibilité.....	54
6.4.2.4	Même objet physique.....	54
6.4.2.5	Concaténation.....	54
6.4.2.6	Alignement de pavés.....	54
6.4.3	Commandes de mises en page applicables en l'absence de structure physique générique.....	54

	<i>Page</i>
6.5	Mise en page du contenu et caractéristiques de restriction 55
6.5.1	Contenu de caractères 55
6.5.1.1	Introduction 55
6.5.1.2	Classes d'architectures de contenu de caractères 56
6.5.1.3	Répertoires de caractères 56
6.5.1.4	Techniques d'extension de code 57
6.5.1.5	Espacement entre les lignes 58
6.5.1.6	Espacement des caractères 58
6.5.1.7	Trajet des caractères et progression des lignes 59
6.5.1.8	Commandes de positionnement des caractères 59
6.5.1.9	Orientation des caractères 59
6.5.1.10	Mise en relief 60
6.5.1.11	Tabulation 60
6.5.1.12	Indentation 61
6.5.1.13	Alignement 61
6.5.1.14	Format de la première ligne 61
6.5.1.15	Taille-veuve et taille-orpheline 61
6.5.1.16	Polices de caractères 62
6.5.1.17	Chaînes de caractères inverses 62
6.5.1.18	Décalage de crénage 62
6.5.1.19	Espacement ligne proportionnel 62
6.5.1.20	Exposants et indices 62
6.5.1.21	Coupure de ligne 62
6.5.1.22	Remplacement de caractères 63
6.5.1.23	Point initial 63
6.5.1.24	Utilisation des fonctions de commande 63
6.5.1.25	Formatage du contenu 64
6.5.1.26	Composition des caractères graphiques 64
6.5.1.27	Textes parallèles 64
6.5.1.28	Crénage couplé 64
6.5.2	Contenu graphique en points 64
6.5.2.1	Introduction 64
6.5.2.2	Architecture de contenu graphique en points 65
6.5.2.3	Méthodes de codage du contenu graphique en points 65
6.5.2.4	Trajet des éléments d'image et progression des lignes 65
6.5.2.5	Découpage 65
6.5.2.6	Espacement des éléments d'image 66
6.5.2.7	Rapport d'espacement 66
6.5.2.8	Dimensions de l'image 66
6.5.3	Contenu graphique géométrique 66
6.6	Caractéristiques diverses 66
6.6.1	Documents ressources 66
6.6.2	Documents externes 67
6.6.3	Frontières 67
6.6.4	Commentaires d'application 67
6.6.5	Représentation de remplacement 67
6.6.6	Mécanismes de numérotation automatique et de référence 67
6.6.6.1	Introduction 67
6.6.6.2	Affectations 69
6.6.6.3	Numérotation de segments 71
6.6.6.4	Numérotation des figures 72
6.6.6.5	Numérotation de listes 72
6.6.6.6	Numérotation de tableaux 72
6.6.6.7	Numérotation de notes de bas de page 73
6.6.6.8	Numérotation des pages 73
6.6.6.9	Contenu référencé 74
6.6.6.10	Références communes 74
6.6.6.11	Références à l'instance courante 75
6.6.6.12	Identificateurs communs 75

6.6.7	Commentaires lisibles par l'utilisateur.....	75
6.6.8	Nom visible par l'utilisateur.....	75
6.6.9	Unit scaling.....	75
6.7	Caractéristiques de gestion du document.....	76
6.7.1	Information sur les constituants du document.....	76
6.7.1.1	Présence des constituants du document.....	76
6.7.1.2	Information sur les documents ressources.....	76
6.7.1.3	Information sur les documents externes.....	76
6.7.2	Information d'identification du document.....	76
6.7.2.1	Information sur le profil d'application de document.....	76
6.7.2.2	Information sur la classe d'architecture de document.....	76
6.7.2.3	Information sur la classe d'architecture de contenu.....	76
6.7.2.4	Information sur la classe de format d'échange.....	76
6.7.2.5	Information sur la version ODA.....	77
6.7.2.6	Référence du document.....	77
6.7.3	Information sur les valeurs par défaut du document.....	77
6.7.4	Information sur les caractéristiques non essentielles.....	77
6.7.4.1	Jeux de caractères de profil.....	77
6.7.4.2	Jeux de caractères de commentaires.....	78
6.7.4.3	Jeux de caractères de représentation de remplacement.....	78
6.7.5	Liste de polices.....	78
6.7.6	Attributs de gestion du document.....	78
7	Spécification des contraintes de constitution.....	79
7.1	Introduction.....	79
7.1.1	Diagrammes des relations des constituants logiques.....	79
7.1.2	Diagrammes des relations des constituants physiques.....	86
7.1.3	Notation.....	92
7.2	Document profile constituant constraints.....	93
7.2.1	Macro definitions.....	93
7.2.2	Constituent constraints.....	98
7.2.2.1	DocumentProfile.....	98
7.3	Logical constituent constraints.....	101
7.3.1	Macro definitions.....	101
7.3.2	Factor constraints.....	115
7.3.2.1	FACTOR ANY-LOGICAL.....	115
7.3.2.2	FACTOR COMP-LOGICAL.....	115
7.3.2.3	FACTOR BASIC-LOGICAL.....	115
7.3.2.4	FACTOR ANY-COMMON.....	116
7.3.3	Constituent constraints.....	116
7.3.3.1	DocumentLogicalRoot.....	116
7.3.3.2	Passage.....	116
7.3.3.3	NumberedSegment.....	117
7.3.3.4	Number.....	117
7.3.3.5	Title.....	117
7.3.3.6	Caption.....	118
7.3.3.7	Paragraph.....	118
7.3.3.8	Phrase.....	118
7.3.3.9	Footnote.....	118
7.3.3.10	FootnoteNumber.....	119
7.3.3.11	FootnoteReference.....	119
7.3.3.12	FootnoteBody.....	119
7.3.3.13	FootnoteText.....	120
7.3.3.14	Figure.....	120
7.3.3.15	BodyText.....	120
7.3.3.16	Reference.....	120
7.3.3.17	ReferencedContent.....	121
7.3.3.18	BodyRaster.....	121
7.3.3.19	BodyGeometric.....	121

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

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

	<i>Page</i>
7.7	Content portion constituant constraints 170
7.7.1	Macro definitions 170
7.7.2	Factor constraints 170
7.7.3	Constituent constraints 171
7.7.3.1	Character-content-portion 171
7.7.3.2	Raster-graphics-content-portion 172
7.7.3.3	Geometric-graphics-content-portion 172
8	Format d'échange 172
8.1	Format d'échange de documents de classe A 172
8.1.1	Format d'échange 172
8.1.2	Identificateur d'objet du profil d'application de document 172
8.1.3	Codage des commentaires d'application 172
8.1.4	Longueur des valeurs de données 173
Annexe A	– Modifications et corrigenda techniques 173
A.1	Modifications 173
A.1.1	Modifications de la norme de base 173
A.1.2	Proposition de modification des normes visant à corriger certains défauts 173
A.2	Corrigenda technique 173
A.2.1	Corrigenda technique à la présente Recommandation 173
A.3	Versions de la Recommandation ODA 174
Annexe B	– Méthodes conseillées 174
B.1	Méthodes de transfert pour l'ODA 174
B.1.1	Acheminement de l'ODA dans les systèmes conformes à la Recommandation X.400 du CCITT (1984) 174
B.1.2	Acheminement de l'ODA sur le FTAM 174
B.1.3	Acheminement de l'ODA sur le DTAM 174
B.1.4	Acheminement de l'ODA sur disquette 175
B.2	Référence aux fontes 175
B.3	Contraintes imposées par la norme ISO 8632 (CGM) dans le cadre du présent profil d'application de document 176
B.3.1	Éléments délimiteurs 176
B.3.2	Éléments descripteurs de métafichier 176
B.3.3	Éléments descripteurs d'image 177
B.3.4	Éléments de commande 177
B.3.5	Éléments primitives graphiques 177
B.3.6	Éléments d'attribut 178
B.3.7	Éléments externes 180
Annexe C	– Bibliographie 180

AVANT-PROPOS

Le présent profil d'application de document a été élaboré en liaison avec plusieurs organisations:

- l'Atelier Asie-Océanie (AOW: Asia Oceania Workshop);
- la Commission d'étude 8 de l'UIT-T;
- l'Atelier Européen sur les Systèmes Ouverts (EWOS: European Workshop for Open Systems);
- l'Atelier des réalisateurs OSE (OIW: OSE Implementors' Workshop).

La liaison entre ces organisations a été opérée durant les réunions du Groupe d'alignement des profils pour l'ODA (PAGODA). Ces réunions ont essentiellement été consacrées à la mise au point d'un unique ensemble de profils d'application de documents ODA alignée au plan international.

La présente Recommandation est propre à l'identificateur de profil PM-36.

Actuellement, la présente Recommandation comporte une partie, à savoir:

- un profil d'application de document.

D'autres parties pourront être ajoutées à la présente Recommandation.

- l'Annexe A (normative): Modifications et corrigenda techniques
- l'Annexe B (informative): Méthodes conseillées
- l'Annexe C (informative): Bibliographie.

INTRODUCTION

L'objet de la présente Recommandation est de faciliter l'interfonctionnement des applications d'échange de documents fondé sur l'architecture de document ouverte (ODA) Rec. de la série T.410 du CCITT | ISO 8613-1. La présente Recommandation est applicable à l'échange de documents sous les formes formatée, retraitable ou formatée retraitable et a été élaborée conformément à la Rec. T.411 du CCITT | ISO 8613-1. Le format de la présente Recommandation est conforme au pro forma et à la notation normalisés définis dans l'Annexe F/T.411.

PROFIL D'APPLICATION DE DOCUMENT PM-36 POUR L'ÉCHANGE DE STRUCTURES DOCUMENTAIRES ÉTENDUES ET DE DOCUMENTS À CONTENU MIXTE SOUS FORMES RETRAITABLE ET FORMATÉE¹⁾

1 Champ d'application

La présente Recommandation spécifie les formats d'échange pour le transfert de documents structurés entre équipements de traitement de documents ou de textes. Ces documents peuvent renfermer un contenu caractères, un contenu graphique en points et un contenu graphique géométrique.

Les documents visés par la présente Recommandation vont des simples documents aux rapports techniques très structurés, articles et documents réalisés par composition typographique du type brochures. La présente Recommandation contient un vaste ensemble de caractéristiques permettant le transfert de documents entre ces systèmes.

La présente Recommandation permet d'échanger des documents sous les formes suivantes:

- a) forme formatée;
- b) forme retraitable;
- c) forme retraitable formatée.

Les niveaux d'architecture définis pour ces trois formes ont des fonctions d'adaptation qui permettent de convertir les formats d'échange d'un document d'une forme retraitable dans une autre forme.

La présente Recommandation est indépendante des procédés mis en œuvre dans un système terminal pour créer, éditer ou reproduire des documents. Il est également indépendant des moyens qui peuvent être utilisés pour le transfert des documents (liaisons de communication ou supports d'information, par exemple).

2 Références normatives

Les Recommandations | Normes internationales ci-après contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions applicables à la présente Recommandation. Les versions mentionnées étaient en vigueur au moment de la parution. Toutes les Recommandations sont sujettes à révision et l'attention des parties concernées par la présente Recommandation est attirée sur les dangers de l'utilisation aveugle de versions plus récentes des Recommandations citées ci-après étant donné que les renvois aux références contenues dans les profils de ces Recommandations peuvent être propres à une version donnée. Les membres de l'ISO et de la CEI tiennent à jour la liste des normes et profils internationaux normalisés en vigueur. Le TSB tient à jour une liste des Recommandations en vigueur à l'UIT-T.

2.1 Recommandations | normes internationales couplées équivalentes par le contenu technique

- Recommandation T.411 du CCITT (1988), *Architecture de transfert de documents ouverte (ODA) et format d'échange – Introduction et principes généraux.*
ISO 8613-1:1989, *Traitement de l'information – Bureautique, Architecture et format de transfert de documents de bureau (ODA) – Partie 1: Introduction et principes généraux.*
- Recommandation T.411 du CCITT (1991), *Architecture de transfert de documents ouverte (ODA) et format d'échange – Introduction et principes généraux – Annexe F: Pro forma et notation correspondante du profil d'application de document.*
ISO 8613-1 Add.1, *Traitement de l'information – Bureautique, Architecture et format de transfert de documents de bureau (ODA) – Partie 1: Introduction et principes généraux – Addendum 1: Pro forma et notation correspondante du profil d'application de document.*

¹⁾ La Recommandation UIT-T T.506 et ISO/CEI ISP 11182.1 ont été élaborées en étroite collaboration et sont techniquement alignées, sauf en ce qui concerne l'utilisation de la notation SGML et le format d'interchange SDIF.

- Recommandation T.412 du CCITT (1988), *Architecture de documents ouverte (ODA) – Structures des documents*.
ISO 8613-2:1989, *Traitement de l'information – Bureautique, Architecture et format de transfert de documents de bureau (ODA) – Partie 2: Structures des documents*.
- Recommandation T.414 du CCITT (1988), *Architecture de documents ouverte (ODA) et format d'échange – Profil de document*.
ISO 8613-4:1989, *Traitement de l'information – Bureautique, Architecture et format de transfert de documents de bureau (ODA) – Partie 4: Profils des documents*.
- Recommandation T.415 du CCITT (1988), *Architecture de documents ouverte (ODA) et format d'échange: Format ouvert d'échange de documents*.
ISO 8613-5:1989, *Traitement de l'information – Bureautique, Architecture et format de transfert de documents de bureau (ODA) – Partie 5: Formats de transfert de documents de bureau*.
- Recommandation T.416 du CCITT (1988), *Architecture de documents ouverte (ODA) et format d'échange – Architecture à contenu caractère*.
ISO 8613-6:1989, *Traitement de l'information – Bureautique, Architecture de documents de bureau (ODA) et format d'échange – Partie 6: Architecture à contenu caractères*.
- Recommandation T.417 du CCITT (1988), *Architecture de documents ouverte (ODA) et format d'échange – Architecture à contenu graphique mode point*.
ISO 8613-7:1989, *Traitement de l'information – Bureautique, Architecture de documents de bureau (ODA) et format d'échange – Partie 7: Architecture à contenu caractères graphiques mode point*.
- Recommandation T.418 du CCITT (1988), *Architecture de documents ouverte (ODA) et format d'échange – Architecture à contenu graphique géométrique*.
ISO 8613-8:1989, *Traitement de l'information – Bureautique, Architecture de documents de bureau (ODA) – Partie 8: Architecture à contenu caractères graphiques mode géométrique*.
- Recommandation X.208 du CCITT (1988), *Recommandation de la syntaxe Abstraite Numéro Un (ASN.1) géométrique*.
ISO/CEI 8824:1990, *Technologies de l'information – Interconnexion des systèmes ouverts – Notation de Syntaxe Abstraite Numéro Un (ASN.1)*.
- Recommandation X.209 du CCITT (1988), *Recommandations des règles de codage de la Notation de Syntaxe Abstraite Numéro Un (ASN.1)*.
ISO/CEI 8825:1990, *Technologies de l'information – Interconnexion des systèmes ouverts – Règles de base pour coder la Notation de Syntaxe Abstraite Numéro Un (ASN.1)*.
- Recommandation T.502 du CCITT (1991), *Profil d'application de document PM-11 d'échange de documents à contenu caractère sous forme formatée retraitable*.
ISO/CEI ISP 10610-1:1992, *Technologies de l'information – Profil normalisé international FOD 011 – Format de document ouvert: structure de document simple – Architecture de contenu de caractères seulement – Profil d'application de document*.
- Recommandation T.505 du CCITT (1991), *Profil d'application de document PM-26 de transfert de documents à contenu mixte sous forme formatée retraitable*.
ISO/CEI ISP 11181-1:1992, *Technologies de l'information – Profil normalisé international FOD 026 – Format de document ouvert: structure de document améliorée – Architecture de contenus de caractères, de graphiques en points et de graphiques géométriques – Profil d'application de document*.

2.2 Références additionnelles

- Recommandation T.4 du CCITT (1988), *Normalisation des télécopieurs du groupe 3 pour la transmission de documents*.
- Recommandation T.6 du CCITT (1988), *Schémas de codage et fonctions de commande de codage de télécopie pour les télécopieurs du groupe 4*.
- Recommandation T.400 du CCITT (1988), *Introduction à l'architecture, au transfert et à la manipulation de documents*.
- ISO 646:1991, *Traitement de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'information*.

- ISO 2022:1986, *Traitement de l'information – Jeux ISO de caractères codés à 7 et 8 éléments – Techniques d'extension de code.*
- ISO 2375:1985, *Traitement de l'information – Procédures pour l'enregistrement des séquences d'échappement.*
- ISO 6937-2:1983, *Traitement de l'information – Jeux de caractères codés pour la transmission de texte – Partie 2: Caractères graphiques alphabétiques latins et caractères graphiques non alphabétiques.*
- ISO 6937-2:1983/Add.1: 1989, *Traitement de l'information – Jeux de caractères codés pour la transmission de texte – Partie 2: Caractères graphiques alphabétiques latins et caractères graphiques non alphabétiques. Addendum 1.*
- ISO/CEI 7350:1990, *Traitement de l'information – Enregistrement des répertoires de caractères graphiques de l'ISO 10367.*
- ISO/CEI 8632:1992, *Traitement de l'information – Infographie – Métafichier pour le stockage et le transfert des informations de description d'image.*
- ISO 8859-1:1987, *Traitement de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1: Alphabet latin n° 1.*
- ISO 8879:1986, *Traitement de l'information – Systèmes bureautiques – Langage normalisé de balisage généralisé (SGML).*
- ISO 9069:1988, *Traitement de l'information – Facilités de support SGML – Format d'échange de documents SGML (SDIF).*
- ISO 9293:1987, *Traitement de l'information – Volume et structure des fichiers des cartouches à disquette pour l'échange d'information.*
- ISO/CEI 9541:1991, *Technologies de l'information – Echange d'informations sur les fontes.*
- ISO/CEI TR 10000-1:1991, *Technologies de l'information – Cadre et taxonomie des Profils normalisés internationaux – Partie 1: Cadre.*
- ISO/CEI TR 10000-2: 1990, *Technologies de l'information – Cadre et taxonomie des Profils normalisés internationaux – Partie 2: Taxonomie.*
- ISO/CEI 11182-1:1993, *Technologies de l'information – Profil normalisé international FOD36 – Format de document ouvert: Structure de document avancé – Architectures du contenu des caractères, des graphiques à raster et des graphiques géométriques – Partie 1: Profil du document d'application (DAP)*

3 Définitions

Pour les besoins de la présente Recommandation, les définitions suivantes sont utilisées.

Les définitions données dans la Recommandation T.411 | Norme ISO 8613-1 sont applicables.

noms de contrainte de constitution: Chaque constituant qui peut être inclus dans un document conforme à la présente Recommandation a reçu un nom unique qui sert à associer ce constituant à une contrainte de constituant définie dans la présente Recommandation.

Par convention, on utilise des noms entiers (c'est-à-dire pas d'abréviations), les deux mots ou plus qui composent un nom étant concaténés et chaque mot commençant par une majuscule. Exemples de noms de contrainte de constitution utilisés dans la présente Recommandation: BodyText, Footnote, RectoPage et ColumnFixed.

A l'article 6 de la présente Recommandation chaque contrainte de constitution fournie dans la présente Recommandation est soulignée une fois à l'endroit du texte où sa fonction est définie ce qui permet par ailleurs d'identifier toutes les contraintes de constitution de la présente Recommandation.

Les mêmes noms de contrainte de constitution sont également utilisés dans la spécification technique de l'article 7 de sorte qu'il existe une correspondance biunivoque entre l'utilisation de ces noms dans les articles 6 et 7.

Les noms des contraintes de constitution se rapportent à leur objet, mais la sémantique ne doit pas être déduite implicitement des noms réels qui sont utilisés. En outre, ces noms n'apparaissent pas dans un document échangé. Toutefois, un mécanisme permettant d'identifier les contraintes de constituant dans un document échangé est prévu (voir 6.6.4). Ainsi, dans une application utilisant la présente Recommandation, les constituants peuvent être connus de l'utilisateur sous différents noms.

4 Relations avec d'autres profils

La présente Recommandation fait partie d'une série de profils qui s'y rattachent hiérarchiquement et qui comprennent le FOD 011 | PM-11 et le FOD 026 | PM-26.

Les caractéristiques admises par la présente Recommandation constituent une version élaborée des caractéristiques admises par les profils FOD 011 | PM-11 et FOD 026 | PM-26. Par conséquent, tous les trains de données conformes aux profils FOD 011 | PM-11 et FOD 026 | PM-26 sont également conformes à la présente Recommandation excepté l'identificateur DAP.

5 Conformité

Pour être conforme à la présente Recommandation, un train de données qui représente un document doit remplir les conditions spécifiées en 5.1.

La présente Recommandation ne définit pas les conditions de mise en œuvre ou de service. Ces conditions sont définies dans d'autres Recommandations qui utilisent la présente Recommandation.

5.1 Conformité de train de données

Les conditions ci-après sont applicables au codage des trains de données conformes à la présente Recommandation:

- a) Le train de données doit être codé conformément aux règles de codage ASN.1 définies dans la Rec. X.209 du CCITT | Norme ISO 8825.
- b) Le train de données doit être structuré conformément au format d'échange défini à l'article 8.
- c) Le document, tel qu'il est représenté dans le train de données après résolution des références externes éventuelles, doit être structuré conformément à l'une des classes d'architectures de document définies en 6.1 et doit contenir tous les constituants obligatoires spécifiés pour cette classe; d'autres constituants peuvent être inclus, à condition qu'ils soient permis pour cette classe, comme indiqué à l'article 7.
- d) Chaque constituant doit contenir tous les attributs spécifiés conformément aux exigences de la présente Recommandation pour ce constituant; d'autres attributs peuvent être spécifiés à condition qu'ils soient permis pour ce constituant.
- e) Les valeurs d'attribut spécifiées doivent être comprises dans la gamme des valeurs admissibles spécifiées dans la présente Recommandation.
- f) Le document codé doit être construit conformément à l'architecture de document abstraite définie dans la Rec. T.412 | Norme ISO 8613-2.
- g) Le document doit être structuré conformément aux caractéristiques et contraintes spécifiées à l'article 6.

5.2 Conformité d'application

La présente section énonce les conditions à remplir pour les mises en œuvre qui demandent la conformité avec la présente Recommandation.

Les caractéristiques d'application associées à la présente Recommandation sont définies dans la Recommandation T.516 à développer.

6 Caractéristiques admises par le profil d'application documentaire

Le présent article spécifie les caractéristiques des documents qui peuvent être représentées par des trains de données conformes à la présente Recommandation. Elle précise aussi comment ces caractéristiques sont représentées en termes de contraintes de constitution.

6.1 Vue d'ensemble

6.1.1 Considérations générales

La présente Recommandation assure l'échange de documents sous les formes suivantes:

- forme retraitable, qui facilite la révision d'un document par le destinataire;
- forme formatée, qui facilite la reproduction d'un document selon les souhaits de l'expéditeur;
- forme retraitable formatée, qui facilite la reproduction d'un document selon les intentions de l'expéditeur, ou la révision d'un document par un destinataire.

De plus, la présente Recommandation permet l'échange de:

- documents génériques;
- profil de document.

Les constituants formant ces cinq classes de trains de données sont définis en 6.1.2 à 6.1.6. Les constituants «obligatoires» doivent être présents dans tout document conforme à la présente Recommandation. Les constituants «facultatifs» ne sont présents que si le train de données considéré l'exige.

Les constituants d'un document complet conforme à la présente Recommandation comprennent tous ceux auxquels font référence et que contiennent, le cas échéant, les documents ressources et les documents externes (voir les 6.6.1 et 6.6.2).

6.1.2 Documents sous forme formatée

Constituants obligatoires:

- profil de document;
- descriptions d'objets physiques représentant une structure physique spécifique.

Constituants facultatifs:

- descriptions de classes d'objets physiques représentant une structure physique générique «mise en facteur»;
- styles de présentation;
- descriptions de parties de contenu.

6.1.3 Documents sous forme retraitable

Constituants obligatoires:

- profil de document;
- descriptions de classes d'objets logiques représentant une structure logique générique complète ou partielle;
- descriptions d'objets logiques, représentant une structure logique spécifique.

Constituants facultatifs:

- descriptions de classes d'objets physiques, représentant une structure physique générique complète;
- styles de mise en page;
- styles de présentation;
- descriptions de parties de contenu.

Dans le cas de documents sous forme retraitable, lorsque la structure physique générique n'est pas présente, des restrictions supplémentaires sont imposées en ce qui concerne les directives de formatage qui peuvent être incluses dans les styles de mise en page. Ces restrictions sont définies en 6.4.3.

6.1.4 Documents sous forme retraitable formatée

Constituants obligatoires:

- profil de document;
- descriptions de classes d'objets logiques, représentant une structure logique générique complète ou partielle;
- descriptions d'objets logiques, représentant une structure logique spécifique;
- descriptions de classes d'objets physiques, représentant une structure physique générique complète;
- descriptions d'objets physiques, représentant une structure physique spécifique.

Constituants facultatifs:

- styles de mise en page;
- styles de présentation;
- descriptions de parties de contenu.

6.1.5 Documents génériques

Un document générique se compose des ensembles de constituants suivants:

a)

- un profil de document;
- des descriptions de classes d'objets logiques qui représentent une structure logique générique complète ou partielle;
- des styles de mise en page dont la présence est facultative;
- des styles de présentation dont la présence est facultative;
- des parties de contenu générique dont la présence est facultative;

b)

- un profil de document;
- des descriptions de classes d'objets de formatage qui représentent une structure de mise en page générique complète ou un ensemble facteur;
- des styles de présentation dont la présence est facultative;
- des parties de contenu générique facultatives;

c)

- un profil de document;
- des descriptions de classes d'objets logiques qui représentent une structure logique générique complète ou partielle;
- des descriptions de classes d'objets physiques qui représentent une structure physique générique complète;
- des styles de mise en page dont la présence est facultative;
- des styles de présentation dont la présence est facultative;
- des parties de contenu générique dont la présence est facultative.

6.1.6 Profil de document

Cette forme de document contient seulement un profil de document.

6.2 Caractéristiques logiques

6.2.1 Introduction

La présente section définit les contraintes de constitution logiques imposées par la présente Recommandation qui permettent de représenter les caractéristiques d'un document contenant des descriptions de composantes logiques.

On peut utiliser diverses contraintes de constitution pour représenter et différencier les parties d'un document ayant des caractéristiques logiques différentes. La présente section décrit les caractéristiques générales et les utilisations types des contraintes de constitution offertes.

Les descriptions des caractéristiques logiques représentées par chacune des contraintes de constitution sont données à titre indicatif; il appartient à l'utilisateur de déterminer comment un document doit être représenté à l'aide des constituants disponibles. Toutefois, l'observation des indications données peut faciliter la compréhension mutuelle d'un document par l'expéditeur et le destinataire.

6.2.2 Vue d'ensemble de la structure logique

Du point de vue logique, le document se compose de deux parties, à savoir un corps et une partie commune.

Le corps, qui constitue le contenu essentiel d'un document, est reproduit dans la zone constituant le corps des pages du document.

La partie commune constitue le contenu commun à placer dans les zones d'en-tête et de bas de page réservées sur chaque page d'un document. Les contenus de l'en-tête et du bas de page sont facultatifs indépendamment l'un de l'autre et peuvent donc être inclus dans un document échangé uniquement si besoin est.

6.2.3 Corps de la structure logique

6.2.3.1 DocumentLogicalRoot (Racine logique de document)

Le niveau supérieur de la structure logique d'un document est représenté par la contrainte *DocumentLogicalRoot*. Ses éléments subordonnés immédiats consistent en une séquence arbitraire d'une ou de plusieurs contraintes de constitution des types *Passage* et *NumberedSegment*.

Les plans de numérotation automatique qui s'appliquent aux constituants des types *NumberedSegment*, *Figure*, *Table* et *Footnote* peuvent être initialisés sur le *DocumentLogicalRoot*.

6.2.3.2 Passage (Passage)

Passage est une contrainte de constitution qui représente une subdivision d'un document correspondant à un groupement logique d'éléments subordonnés de ce document. Ce groupement peut être considéré comme une entité logique pour la lecture ou représenter des parties d'un document qui possède des caractéristiques de mise en page et de présentation communes.

Les passages servent généralement à représenter:

- le sommaire à placer sur la page de titre d'un rapport;
- le premier sujet traité dans l'avant-propos ou la table des matières;
- le sujet principal du document;
- le dernier sujet traité, constituant les appendices, le glossaire et l'index;
- une figure assortie d'un texte qui est inséré comme entité distincte dans une section du document.

Les plans de numérotation automatique qui s'appliquent aux constituants subordonnés des types *NumberedSegment*, *Figure*, *Table* et *Footnote* peuvent être initialisés sur un *Passage*.

Les éléments subordonnés immédiats d'un *Passage* se composent d'un constituant facultatif du type *Title*, suivi d'une séquence arbitraire facultative d'un ou de plusieurs des types de constituants suivants:

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

Un *Passage* doit au moins avoir un des types de constituants ci-dessus comme élément subordonné.

Il ressort de ce qui précède qu'un *Passage* peut lui-même renfermer un ou plusieurs éléments subordonnés du type *Passage* et qu'en conséquence, les *Passages* peuvent être emboîtés dans une séquence de niveaux. Ainsi, les entités logiques d'un document peuvent être décrites par leurs composants. Un *NumberedSegment* peut aussi contenir un ou plusieurs *Passages* subordonnés, ce qui permet de différencier les diverses entités logiques qu'il renferme.

Un document peut contenir différentes définitions de classes de type *Passage* qui indiquent les caractéristiques communes d'ensembles de *Passages* à l'intérieur du document tels que leurs éléments subordonnés autorisés ou les propriétés de mise en page. Par exemple, on peut définir une classe de *Passages* qui commence toujours sur un nouvel ensemble de pages.

6.2.3.3 NumberedSegment (Segment numéroté)

NumberedSegment est une contrainte de constitution qui représente une entité logique d'un document, qui est différenciée par un identificateur. Il peut s'agir de la subdivision d'un document ou d'un Passage ou d'un *NumberedSegment* de niveau supérieur. On peut aussi différencier ces entités les unes des autres en leur attribuant des caractéristiques physiques communes.

Les systèmes de numérotation automatique qui s'appliquent aux constituants subordonnés des types *NumberedSegment*, *Figure*, *Table*, *NumberedList* et *Footnote* peuvent être initialisés sur un objet logique ou une classe d'objet logique, typiquement sur un *Passage* ou un *NumberedSegment*.

Les éléments subordonnés immédiats d'un *NumberedSegment* se composent du constituant *Number*, dont la présence est obligatoire et qui sert à acheminer l'identificateur du *NumberedSegment*. Ce dernier est suivi, le cas échéant, d'un constituant du type *Title*, qui est suivi à son tour d'une séquence facultative arbitraire d'un ou de plusieurs des types de contraintes de constitution suivantes:

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

En conséquence, un document peut contenir un nombre quelconque de niveaux emboîtés du constituant *NumberedSegment*.

Un *NumberedSegment* est généralement utilisé pour représenter des entités telles que chapitres, sections, sous-sections emboîtées et appendices, qui contiennent un identificateur qui sert à les différencier les unes des autres, ce qui facilite la compréhension pour les personnes intéressées.

Un document peut contenir différentes définitions de classes de constituant *NumberedSegment* qui indiquent les caractéristiques communes d'ensembles de *NumberedSegment*, telles que leurs éléments subordonnés autorisés et les propriétés de mise en page.

Les définitions de classe de *NumberedSegment* ne peuvent pas être définies de façon récurrente. Dans ce cas, seul une classe *NumberedSegment* peut être spécifiée, et la construction `<simple-expr>` dans la macro `USENUMBEREDSTRING` dans les attributs liants de cette classe doivent utiliser uniquement la construction `ORD` facultative. Si l'on utilise des définitions de classe récursives pour le *NumberedSegment*, les contraintes ci-après doivent également être appliquées. Pour tous les niveaux qui désignent des classes définies de manière récursive:

- le format de numérotation doit être le même;
- aucune valeur initiale non égale à 1 ou une réinitialisation du numérotage est possible;
- il n'est pas possible de continuer le numérotage à travers les Passages.

6.2.3.4 Number (Numéro)

Number est une contrainte de constitution qui représente l'identificateur d'un *NumberedSegment*, d'une *Figure* ou d'une *NumberedList* auquel il est subordonné. Cet identificateur permet de différencier à l'intérieur du document le constituant supérieur dont il relève pour le traitement machine ou pour faciliter la compréhension pour les personnes.

Un numéro est un constituant logique de base qui contient un générateur de contenu qui, une fois évalué, produit l'identificateur susmentionné. Cette évaluation a lieu pendant le processus de mise en page.

Les identificateurs sont structurés et consistent en une séquence d'un ou de plusieurs chiffres permettant de distinguer de manière unique les *NumberedSegments* de même niveau ou de niveaux différents dans la structure d'un document. Les chiffres peuvent être représentés par des chiffres arabes ou romains ou par les lettres de l'alphabet correspondantes en caractères minuscules ou majuscules (le numéro 1 est représenté par A, etc.). Les numéros d'un identificateur peuvent être distingués à l'aide de caractères «séparateurs» tels que les espaces et les points; un exemple type est «6.2.3.4».

NOTE – Le séparateur peut être une chaîne vide.

On trouvera en 6.6.6 des précisions sur la structure et la génération des identificateurs.

6.2.3.5 Paragraph (Paragraphe)

Paragraph est une contrainte de constitution qui est une subdivision d'un Passage ou d'un NumberedSegment. Il est généralement utilisé pour représenter le groupement de parties d'un document qui traite d'un thème ou d'un sujet unique. Ces parties peuvent se composer d'un contenu caractères, d'un contenu graphique en points et d'un contenu graphique géométrique.

Les éléments subordonnés immédiats de Paragraph consistent en une séquence arbitraire d'un ou de plusieurs des types de constituants suivants:

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

Le contenu caractères associé à Paragraph peut être «concaténé» pour former un flot ininterrompu de contenu caractères. Celui-ci peut comporter une seule ou plusieurs références à des notes de bas de page ou à d'autres parties d'un document et peut se présenter sous la forme d'une unité unique. Les entités de ce contenu ayant une signification logique ou des caractéristiques de présentation particulières peuvent être différenciées à l'aide du constituant phrase.

Le contenu des éléments subordonnés d'un paragraphe peuvent être séparés entre eux pour offrir un espace entre eux en utilisant Separation (voir 6.4.2.2). On peut utiliser ce moyen pour donner un effet analogue à celui qu'on obtient avec des lignes de texte vides. L'utilisation de lignes de texte vides pour obtenir un espace entre plusieurs zones de texte ou plusieurs autres contenus peut se traduire par des zones blanches non souhaitées adjacentes au bord d'attaque d'objets de mise en page (coupures de page par exemple) ce que l'utilisation de Separation permet d'éviter.

6.2.3.6 BodyText, BodyRaster et BodyGeometric (corps de texte, contenu graphique en points de corps de texte et contenu graphique géométrique de corps de texte)

BodyText, *BodyRaster* et *BodyGeometric* sont des contraintes de constitution qui représentent le niveau le plus bas de subdivision logique d'un document. Ces constituants servent à acheminer le contenu du document et peuvent être spécifiés comme étant des éléments subordonnés de contraintes de constitution :

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

En outre, BodyText peut être spécifié comme élément subordonné à Phrase, Caption et Description. Ces contraintes de constitution permettent de spécifier les caractéristiques de mise en page et de présentation des différentes parties du contenu du document.

Il s'agit de contraintes de constitution logiques de base qui se rapportent directement aux parties de contenu qui contiennent respectivement le contenu caractères, graphique en points et graphique géométrique. BodyText peut renvoyer à une ou plusieurs parties de contenu contenant chacune un contenu caractères retraitsable, formaté ou retraitsable formaté. BodyRaster et BodyGeometric ne peuvent renvoyer qu'à une seule partie de contenu contenant un contenu graphique en points retraitsable formaté ou un contenu graphique géométrique retraitsable formaté, respectivement.

Les contraintes de constitution de ces types dans la structure logique générique peuvent se rapporter à des contenus génériques. Cela permet de définir le contenu commun du corps d'un document.

Lorsque la contrainte de constitution supérieure en question est un élément subordonné à un `FootnoteBody`, il faut spécifier un des noms de catégorie de mise en page pour cette contrainte de constitution, 'Footnote' ou 'Footnote-<n>'. En associant à ce nom un attribut de 'catégorie autorisée' du même nom sur le cadre de la note de bas de page, on s'assure qu'un objet logique provenant de cette contrainte de constitution apparaît dans un cadre `FootnoteArea` lorsqu'une structure de mise en page générique est spécifiée à l'intérieur du document.

6.2.3.7 Footnote (Note de bas de page)

Footnote est une contrainte de constitution qui sert à représenter des notes de bas de page dans un document. Ce constituant peut être spécifié comme étant un élément subordonné de:

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

Une note de bas de page est un contenu qui est logiquement associé à une partie donnée du corps du document mais qui est destinée à être lue et formatée indépendamment de la partie du document auquel elle se rapporte. En général, une note de bas de page se compose d'un identificateur de note de bas de page, qui est inséré dans le corps du document et de la note de bas de page proprement dite, qui est formatée ailleurs.

Une note de bas de page est un constituant logique composite dont les éléments subordonnés immédiats se composent du constituant `FootnoteReference`, qui représente l'identificateur de note de bas de page, suivi du constituant `FootnoteBody`, qui représente la note de bas de page. Ces deux éléments subordonnés sont obligatoires.

6.2.3.8 FootnoteReference (Renvoi à une note de bas de page)

FootnoteReference est une contrainte de constitution qui sert à représenter un renvoi à une note de bas de page dans le corps d'un document.

`FootnoteReference` est un constituant logique de base qui contient un générateur de contenu qui, une fois évalué, produit une chaîne de caractères qui constitue le renvoi à la note de bas de page susmentionné.

Cette chaîne de caractères produite consiste en une étiquette avec des chaînes de caractères de préfixe et de suffixe facultatifs. L'étiquette sert à identifier une note de bas de page; elle peut consister en un nombre représenté sous forme de chiffres romains ou arabes ou par un équivalent alphabétique. Le nombre peut être produit automatiquement de manière que sa valeur soit augmentée d'une unité à mesure que se suivent les notes de bas de page. L'étiquette peut aussi se composer d'une chaîne de caractères définie par l'utilisateur.

Dans une séquence de notes de bas de page, on peut librement associer la numérotation automatique et des étiquettes définies par l'utilisateur (donnant par exemple la séquence 1,2,*,3,4,). Si l'étiquette consiste en une chaîne de caractères définie par l'utilisateur, la séquence à numérotation automatique n'est pas incrémentée.

Exemple de renvoi à une note de bas de page: «(2)», où respectivement «(et)» sont des chaînes de préfixe et de suffixe définies par l'utilisateur et où «2» est l'étiquette produite automatiquement. Autre exemple: «note⁵», où «5» est l'étiquette et «note» est une chaîne de préfixes qui contient aussi la fonction de contrôle PLU pour permettre à l'étiquette d'être représentée sous forme d'exposant. Dans ce cas, une chaîne de suffixes, contenant la fonction de commande PLD sera nécessaire pour annuler la mise en exposant avant le texte suivant.

Le format du générateur de contenu susmentionné est décrit en 6.6.6.7.

6.2.3.9 FootnoteBody (Corps de texte de note de bas de page)

FootnoteBody est une contrainte de constitution qui représente le contenu d'une note de bas de page. Celui-ci est constitué d'un train de données de contenu caractères qui peut renfermer des références intercalées à d'autres parties du document.

`FootnoteBody` est un constituant logique composite dont les éléments subordonnés se composent du constituant `FootnoteNumber`, qui est obligatoire et représente l'identificateur de note de bas de page, suivi d'une séquence d'un ou de plusieurs constituants des types `FootnoteText` et `Reference`, qui représentent le contenu de la note de bas de page.

L'identificateur susmentionné est identique à l'identificateur de note de bas de page correspondant qui est incorporé au contenu du corps du document et représenté par le constituant FootnoteReference.

Les contraintes de constitution subordonnées à FootnoteBody sont destinées à être mises en page indépendamment des autres parties du contenu du document. Quand une structure physique générique est spécifiée pour le document, ses constituants sont obligatoirement mis en page dans un cadre FootnoteArea (voir le 6.3.5.9).

6.2.3.10 FootnoteNumber (Numéro de note de bas de page)

FootnoteNumber est une contrainte de constitution qui représente l'identificateur de note de bas de page dans le corps de la note de bas de page.

Cet identificateur est identique au contenu associé au constituant FootnoteReference, mais il est destiné à être disposé de manière qu'il précède immédiatement le contenu du corps de la note de bas de page.

FootnoteNumber est un constituant logique de base qui contient un générateur de contenu qui, une fois évalué, produit l'identificateur susmentionné. Le format de ce générateur de contenu est le même que le générateur de contenu qui peut être spécifié pour le constituant Footnotereference.

Il faut spécifier un des noms de catégorie de mise en page pour cette contrainte de constitution, 'Footnote' ou 'footnote-<n>'. En associant à ce nom un attribut de «catégorie autorisée» du même nom sur le cadre de la note de bas de page, on s'assure qu'un objet logique provenant de cette contrainte de constitution apparaît dans un cadre FootnoteArea lorsqu'une structure de mise en page générique est spécifiée à l'intérieur du document.

6.2.3.11 FootnoteText (Texte de note de bas de page)

FootnoteText est une contrainte de constitution qui sert à représenter le contenu de la note de bas de page. C'est la plus petite subdivision logique d'un FootnoteBody.

FootnoteText est un constituant logique de base qui se rapporte à une ou plusieurs parties de contenu contenant chacune un contenu de caractères retraitable, formaté ou retraitable formaté.

Il faut spécifier un des noms de catégorie de mise en page pour cette contrainte de constitution, 'Footnote' ou 'footnote-<n>'. En associant à ce nom un attribut de «catégorie autorisée» du même nom sur le cadre de la note de bas de page, on s'assure qu'un objet logique provenant de cette contrainte de constitution apparaît dans un cadre FootnoteArea lorsqu'une structure de mise en page générique est spécifiée à l'intérieur du document.

6.2.3.12 Title (Titre)

Title est une contrainte de constitution qui sert à représenter le titre, l'en-tête ou le nom du Passage ou du NumberedSegment dont il est le subordonné immédiat. Ce constituant peut se composer d'un contenu caractères, d'un contenu graphique en points et d'un contenu graphique géométrique.

Les éléments subordonnés immédiats de ce constituant se composent d'une séquence arbitraire d'un ou de plusieurs des types de contraintes de constitution suivantes:

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

Le contenu caractères associé à Title peut être concaténé pour former un flot ininterrompu de contenu caractères. Celui-ci peut comporter une seule ou plusieurs références à des notes de bas de pages ou à d'autres parties du document et peut se présenter sous la forme d'une unité unique. Les entités de ce contenu qui ont une signification logique ou des caractéristiques de présentation particulières peuvent être différenciées à l'aide de la contrainte sur les constituants Phrase. Le contenu provenant d'un objet de texte de base avec éléments subordonnés dans un paragraphe peut contenir sur la même ligne en utilisant Concaténation (voir 6.4.2.5). Ou bien, le contenu provenant des éléments subordonnés d'un paragraphe peuvent être séparés entre eux pour obtenir un espace blanc entre eux, en utilisant Separation (voir 6.4.2.2). Cela peut être utilisé pour obtenir un effet analogue à celui obtenu avec des lignes vides de texte.

L'utilisation de lignes vides de texte pour obtenir un espace entre des zones de texte ou autres contenus peut se traduire par la présence de zones blanches intempestives sur le bord d'attaque des objets de mise en page, ce qui ne se produit pas lorsqu'on utilise Separation.

6.2.3.13 Phrase (Phrase)

Phrase est une contrainte de constitution qui sert à regrouper une partie de contenu caractères représentant une seule entité logique qu'il convient de différencier dans un but particulier. Autrement dit, le contenu représenté par une Phrase peut avoir une signification logique particulière ou exiger certaines caractéristiques de mise en page ou de présentation. Le contenu caractères peut renfermer des notes de bas de page intercalées et des références à d'autres parties du contenu du document, par exemple une citation à reproduire en italiques.

Une Phrase peut être utilisée pour subdiviser les constituants des types Paragraph, Title, Caption, Description, Artwork et ListItem. Une Phrase peut aussi être subordonnée à une autre Phrase, ce qui permet d'emboîter les constituants du type Phrase.

Les éléments subordonnés immédiats de ce constituant se composent d'une séquence arbitraire d'un ou de plusieurs des types de contraintes de constitution suivantes:

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

Le contenu caractères associé à une Phrase peut être concaténé pour former un flot ininterrompu de contenu caractères. Celui-ci peut comporter une seule ou plusieurs références à des notes de bas de page ou à d'autres parties du document et peut se présenter sous la forme d'une unité unique. Le contenu caractères peut aussi renfermer des commandes de nouvelles lignes permanentes qui permettront de séparer des parties du contenu lors de la mise en page.

6.2.3.14 Constituants prévoyant un mécanisme général de référence

Le présent paragraphe définit les contraintes de constitution qui permettent de disposer d'un mécanisme général de renvoi dans un document.

6.2.3.14.1 Reference (Référence)

Reference est une contrainte de constitution représentant une référence composée d'un contenu caractères qui est dérivé entièrement ou partiellement d'une autre partie du document. Ce constituant fournit un mécanisme général de référence dans un document.

La contrainte sur le constituant peut être spécifiée comme étant un élément subordonné des contraintes de constitution:

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

Il s'agit d'un constituant composite dont les éléments subordonnés immédiats peuvent former une séquence de contraintes de constitution des types BodyText, ReferencedContent et BodyText.

Le format général du contenu associé au constituant du type Reference est le suivant:

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

Les chaînes de préfixes et de suffixes sont facultatives et sont représentées, au besoin, par un constituant du type BodyText. La chaîne de références est représentée par le constituant ReferencedContent.

Ainsi, une chaîne de références peut faire référence à un identificateur, par exemple un numéro qui différencie un chapitre ou une section, un tableau, une figure, une note de bas de page, un élément d'une liste numérotée ou un numéro de page. Chaque référence peut comporter plusieurs références à différentes parties du document. Comme exemple type,

citons la référence «voir le Tableau 3 du Chapitre 2 à la page 4», où les valeurs «3», «2» et «4» sont dérivées automatiquement du tableau, du chapitre et de la page appropriés du document.

6.2.3.14.2 ReferencedContent (Contenu référencé)

ReferencedContent est une contrainte de constitution qui représente une chaîne de caractères contenant une ou plusieurs références au contenu d'autres parties du document (voir 6.2.3.14.1).

Il s'agit d'une contrainte de constitution logique de base qui est immédiatement subordonnée à la contrainte de constitution *Reference*. *ReferencedContent* contient un générateur de contenu qui, une fois évalué, produit la chaîne de caractères renfermant le contenu référencé.

Une séquence de deux constituants de ce type, ou plus, peut être utilisée pour représenter une chaîne de référence composite telle que «voir Tableau 2 section 3.1, commençant p. 6», où les chaînes «2», «3.1» et «6» sont générées automatiquement par référence aux chaînes numériques attachées aux diverses parties du document.

Le format de ce générateur de contenu est décrit en 6.6.6.9.

Lorsque la contrainte supérieure sur les constituants référencée est un élément subordonné à *Footnote Body*, il faut spécifier un des noms de catégorie de mise en page pour ladite contrainte de constitution, 'footnote' ou 'footnote-<n>'. En associant à ce nom un attribut de «catégorie autorisée» du même nom sur le cadre de la note de bas de page, on s'assure qu'un objet logique provenant de cette contrainte de constitution apparaît dans un cadre *FootnoteArea* lorsqu'une structure de mise en page générique est spécifiée à l'intérieur du document.

6.2.3.15 Constituants représentant des illustrations

6.2.3.15.1 Introduction

La présente Recommandation permet de représenter des illustrations ou des figures qui sont des dessins ou des structures organisées simples dont l'exemple type est le formulaire.

Les dessins – en général des diagrammes ou des figures – sont des images avec un seul type de contenu ou résultant de la superposition de deux ou plusieurs images distinctes qui renferment un contenu caractères, un contenu graphique en points ou un contenu graphique géométrique.

Un formulaire se compose généralement d'un ensemble d'entités logiques ayant chacune une signification logique précise. Chacune de ces entités peut être subdivisée en entités logiques subordonnées. Exemple type: un bon de commande contenant un numéro de référence, des informations sur l'expéditeur (nom, adresse et numéro de téléphone) une liste des articles demandés et les dates de livraison prévues.

Les entités structurées qui entrent dans la composition d'un formulaire seraient présentées dans une zone désignée, elle-même subdivisée en zones spécialement réservées pour chaque type d'entité donné. Cette zone désignée est spécifiée par le cadre *FormArea* lorsque la structure physique générique figure dans le document.

Une illustration peut contenir, le cas échéant, un identificateur qui peut servir à la différencier d'autres parties du document, une légende qu'il est possible d'utiliser pour identifier l'objet de l'illustration, et une partie de texte descriptif auquel elle se rapporte.

On trouvera des précisions sur la mise en page des illustrations contenant des dessins ou des structures organisées en 6.4.1.3.7.

Les contraintes de constitution applicables à la représentation des illustrations sont définies ci-dessous.

6.2.3.15.2 Figure (Figure)

Figure est une contrainte de constitution qui sert en général à représenter une illustration. Celle-ci peut être constituée d'un dessin ou d'une structure organisée, comme cela est indiqué en 6.2.3.15.1.

Une *Figure* est un constituant logique composite qu'il est possible de spécifier comme étant un élément subordonné d'un *Passage* ou d'un *NumberedSegment*.

Les éléments subordonnés d'une *Figure* peuvent se composer d'une séquence, des types de constituants suivants, en ordre indifférent:

- Artwork ou Form;
- Number;
- Caption;
- Description.

Un constituant du type Artwork ou Form doit toujours être présent comme élément subordonné du constituant Figure. Les constituants des types Number, Caption et Description sont, indépendamment facultatifs l'un de l'autre. Ils peuvent être présents dans n'importe quel ordre.

6.2.3.15.3 Artwork (Dessin)

Artwork est une contrainte de constitution qui sert généralement à représenter une image graphique à l'intérieur d'une illustration ou d'une figure. Il peut s'agir d'une image simple, représentée par un contenu caractères, un contenu graphique en points ou un contenu graphique géométrique, ou d'une image composite constituée d'un ensemble de ces types de contenu.

Cette contrainte de constitution ne peut être présente que comme élément subordonné d'une contrainte de constitution du type Figure.

Les éléments subordonnés immédiats de Artwork se composent d'une séquence arbitraire d'un ou de plusieurs constituants des types suivants:

- Phrase;
- BodyRaster;
- BodyGeometric.

La contrainte de constitution Phrase est utilisé pour permettre à l'image de renfermer un contenu caractères avec des références à des notes de bas de page ou à d'autres parties du document.

6.2.3.15.4 Form (Formulaire)

Form est une contrainte de constitution qui sert à représenter une forme simple à l'intérieur d'une illustration ou d'une figure. Il s'agit d'un constituant logique composite qui ne peut être présent que comme élément subordonné d'un constituant du type Figure.

Un élément de type Form se compose d'une séquence arbitraire d'entrées de base et d'entrées composites. Les entrées de base, qui peuvent être constituées d'un contenu caractères, d'un contenu graphique en points ou d'un contenu graphique géométrique, sont représentées par des constituants subordonnés du type EntryElement. Les entrées composites sont représentées par des constituants subordonnés du type EntryGroup.

Les contraintes de constitution du type EntryGroup peuvent être subdivisées en un ensemble d'une ou de plusieurs entrées de base ou entrées composites. Ainsi, une entrée composite peut être emboîtée dans une séquence de niveaux telle que chaque niveau se compose d'un ensemble d'entrées de base et d'entrées composites subordonnées.

La présente Recommandation ne prévoit pas de mécanisme permettant de définir la sémantique correspondant à chaque entrée de base ou composite. Une telle définition pourra être donnée, dans l'application considérée, au moyen du paramètre «données externes» de l'attribut «commentaires d'application» (voir 6.6.4).

6.2.3.15.5 Caption (Légende)

Caption est une contrainte de constitution qui représente en général un titre ou un en-tête associé à une illustration ou à une figure. Ce constituant représente un contenu caractères qui peut contenir des références intercalées à des notes de bas de page et à un autre contenu du document.

Une légende est une contrainte de constitution logique composite qui ne peut être présent que comme élément subordonné d'une contrainte de constitution du type Figure.

Les éléments subordonnés immédiats d'une légende se composent d'une séquence arbitraire d'une ou de plusieurs contraintes de constitution des types suivants:

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

Ces contraintes de constitution peuvent être concaténées pour former un flot ininterrompu de contenu caractères, qui doit se présenter sous la forme d'une unité unique.

6.2.3.15.6 Description (Description)

Description est une contrainte de constitution qui représente en général certaines informations supplémentaires à caractère général, qui font partie d'une illustration contenant une figure ou une forme.

Il s'agit d'une contrainte de constitution logique composite qui ne peut être présente que comme élément subordonné d'une contrainte de constitution du type Figure.

Les éléments subordonnés immédiats de Description se composent d'une séquence arbitraire d'une ou de plusieurs contraintes de constitution des types suivants:

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

Ces contraintes de constitution peuvent être concaténées pour former un flot ininterrompu de contenu caractères, qui doit se présenter sous la forme d'une unité unique.

6.2.3.15.7 EntryGroup (Groupe d'entrées)

EntryGroup est une contrainte de constitution qui représente une entrée logique composite dans un formulaire.

Il s'agit d'une contrainte de constitution logique composite qui ne peut être présente que comme élément subordonné de contraintes de constitution des types Form ou EntryGroup.

Les éléments subordonnés immédiats de EntryGroup se composent d'une séquence arbitraire d'une ou de plusieurs contraintes de constitution des types EntryElement et EntryGroup. Les éléments subordonnés d'une contrainte de constitution du type EntryGroup peuvent donc être intégrés dans une séquence de n niveaux.

NOTE – Les constituants du type EntryElement sont définis au 6.2.3.16.6.

6.2.3.16 Constituants utilisés pour représenter des tableaux

6.2.3.16.1 Introduction

Pour les besoins de la présente Recommandation, un tableau désigne une entité logique qui se compose d'une séquence ordonnée d'éléments, appelés cellules, qui sont disposés dans un ensemble à deux dimensions de lignes et de colonnes.

Une ligne peut être une ligne simple contenant une séquence d'une ou de plusieurs cellules. Une ligne peut aussi être composite, c'est-à-dire être composée d'une seule cellule, suivie d'une séquence d'une ou de plusieurs sous-lignes contenant chacune une séquence de cellules.

On trouvera ci-après les constituants logiques utilisés pour représenter des tableaux. La Figure 28 illustre les relations structurelles entre les constituants employés pour représenter un tableau. Le paragraphe 6.4.1.3.8 montre comment les tableaux doivent être mis en page.

6.2.3.16.2 Table (Tableau)

Table est une contrainte de constitution logique qui représente un tableau dans son ensemble. Ce constituant peut être spécifié comme étant un élément subordonné des constituants des types Passage et NumberedSegment.

Les éléments subordonnés immédiats de ce constituant se composent d'une séquence de contraintes de constitution du type Row. Chaque ligne peut avoir ou non les mêmes caractéristiques de sous-structure.

6.2.3.16.3 Row (Ligne)

Row est une contrainte de constitution subordonnée à la contrainte de constitution Table, qui représente une ligne d'entrées dans un tableau; une ligne peut être une séquence d'entrées ou une seule entrée suivie d'une séquence de sous-lignes d'entrées.

Pour représenter ces deux cas, les subordonnés immédiats d'une Row peuvent être:

- une séquence de contrainte de constitution de type EntryElement; ou
- une contrainte de constitution unique du type EntryElement suivi d'une seule contrainte de constitution de type TableComponent.

6.2.3.16.4 TableComponent (Elément de tableau)

TableComponent est une contrainte de constitution au constituant, subordonnée à la contrainte de constitution Row, qui représente une séquence d'une ou plusieurs sous-lignes d'entrées dans une ligne de tableau.

Les subordonnés immédiats de cette contrainte de constitution se composent d'une séquence d'une ou plusieurs contraintes de constitution du type RowComponent. Chaque RowComponent peut, mais ce n'est pas nécessaire, avoir les mêmes caractéristiques de sous-structure.

6.2.3.16.5 RowComponent (Elément de ligne)

RowComponent est une contrainte de constitution, subordonnée à la contrainte de constitution TableComponent, qui représente une sous-ligne d'entrées dans une ligne d'entrées de tableau.

Les subordonnés immédiats de cette contrainte de constitution se composent d'une séquence d'un ou plusieurs composants de type EntryElement. Chaque EntryElement peut, mais ce n'est pas nécessaire, avoir les mêmes caractéristiques de sous-structure.

6.2.3.16.6 EntryElement (Elément d'entrée)

EntryElement est une contrainte de constitution qui représente une seule entrée dans un formulaire ou dans un tableau. EntryElement est une subdivision des types Table ou Form. Dans le cas de Table, elle est spécifiée comme étant subordonnée à une ligne Row ou à un élément de ligne RowComponent, qui représente une seule entrée de tableau. Dans le cas de Form, elle est spécifiée comme étant subordonnée à Form lui-même ou à EntryGroup.

Chaque entrée peut être un contenu caractères, un contenu graphique en points ou un contenu graphique géométrique, si bien que l'élément EntryElement a une seule contrainte de constitution immédiatement subordonnée de type BodyText, BodyRaster ou BodyGeometric.

6.2.3.16.7 EntryText, EntryRaster et EntryGeometric (Texte d'entrée, graphique en points d'entrée et graphique géométrique d'entrée)

EntryText, *EntryRaster* et *EntryGeometric* sont des contraintes de constitution qui représentent un contenu à insérer dans des tableaux et dans des formulaires. Ces contraintes de constitution peuvent être spécifiées comme étant subordonnées à la contrainte de constitution EntryElement et permettent de spécifier les caractéristiques de mise en page et de présentation.

Ce sont des contraintes de base sur les constituants qui se réfèrent directement à des parties contenu contenant des caractères, des graphiques en points et des graphiques géométriques. EntryText se réfère uniquement à une ou plusieurs parties de contenu pouvant contenir un contenu caractère retraitable, formaté ou retraitable formaté. EntryRaster et EntryGeometric se réfèrent uniquement à une seule partie du contenu contenant un contenu graphique en points formaté traitable ou graphique géométrique respectivement.

Les contraintes de constitution de ces types dans la structure logique générique peuvent se référer au contenu logique. C'est un moyen de définir le contenu commun dans les tableaux et les formulaires.

6.2.3.17 Constituants représentant des listes

6.2.3.17.1 Introduction

La présente Recommandation admet trois types de listes:

- listes ordonnées, énumérant des éléments, précédés chacun d'un identificateur (caractères alphabétiques ou chiffres);
- listes non ordonnées, regroupant des éléments, pouvant chacun, à titre facultatif, être précédé d'un caractère de séparation (trait d'union ou petit cercle);
- listes de définitions, composées de paires ordonnées d'éléments (par exemple, terme et définition correspondante).

Toute liste peut être incorporée dans une séquence sans restriction, et une liste peut se composer de listes d'autres types. Par exemple, un élément de liste ordonnée peut être une liste ordonnée subordonnée, une liste non ordonnée ou une liste de définitions.

Les contraintes de constitution pouvant être utilisées pour représenter ces types de listes sont définies ci-après.

6.2.3.17.2 NumberedList (Liste ordonnée)

NumberedList est une contrainte de constitution qui sert à représenter une liste d'éléments précédés chacun d'un identificateur ayant pour fonction de distinguer cet élément.

Cette contrainte de constitution peut être spécifiquement subordonnée immédiatement aux éléments suivants: Passage, NumberedSegment, ListItem, NumberedList, UnNumberedList ou DefinitionList.

Les subordonnés immédiats d'une NumberedList sont la contrainte de constitution Number, laquelle contient le générateur qui génère l'identificateur correspondant à chaque élément dans la liste, suivie par la contrainte de constitution ListItem. Cette paire de contrainte de constitution peut être répétée sans limitation.

De plus amples renseignements concernant la numérotation de listes se trouvent au 6.6.6.5.

6.2.3.17.3 UnNumberedList (Liste non ordonnée)

UnNumberedList est une contrainte de constitution qui sert à représenter une liste d'éléments, pouvant être chacun précédé d'un caractère de séparation facultatif – contenus caractères, contenus graphiques en points ou contenus graphiques géométriques.

Cette contrainte des constituants peut être spécifiée sous forme de subordonné immédiat à Passage, NumberedSegment ou ListItem.

Les subordonnés immédiats d'une UnNumberedList sont le caractère de séparation, représenté par un constituant de type BodyText, RasterGraphics ou GeometricGraphics, suivi de la contrainte de constitution ListItem. Cette paire de contrainte de constitution peut être répétée sans limitation.

6.2.3.17.4 DefinitionList (Liste de définitions)

DefinitionList est une contrainte de constitution qui représente une séquence de paire d'éléments ordonnés.

Ce constituant peut être spécifié sous forme de subordonné immédiat à Passage, NumberedSegment, ListItem.

Les subordonnés immédiats de cet élément sont la contrainte de constitution ListTerm suivi de la contrainte de constitution ListItem. Cette paire de contrainte de constitution peut être répétée sans limitation.

6.2.3.17.5 ListItem (Elément de liste)

ListItem est une contrainte de constitution qui représente un élément de NumberedList, UnNumberedList ou DefinitionList: cette contrainte de constitution représente le second élément de chaque paire d'éléments d'une liste ordonnée, d'une liste non ordonnée ou d'une liste de définitions.

Les subordonnés immédiats de cette contrainte de constitution peuvent être une séquence d'une ou plusieurs contraintes de constitution Phrase ou une contrainte de constitution NumberedList, UnNumberedList ou DefinitionList.

En conséquence, cette contrainte de constitution représente un contenu caractères qui peut contenir des références incorporées à une note de bas de page et à d'autres parties du document, ou représenter une liste d'éléments subordonnés.

6.2.3.17.6 ListTerm (Terme de liste)

ListTerm est une contrainte de constitution qui représente un élément terme d'une DefinitionList. L'élément-terme est le premier élément d'une paire d'éléments dans une liste de définitions.

Les subordonnés immédiats de cette contrainte de constitution sont une séquence ordonnée de façon arbitraire d'un ou plusieurs types de constituants:

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

Cette contrainte de constitution représente donc une quantité de caractères, de graphismes en point et de graphismes géométriques. Le contenu caractères peut comprendre des phrases, références et renvois intégrés.

6.2.4 Contenu commun de la structure logique

6.2.4.1 CommonContent (Contenu commun)

CommonContent est une contrainte de constitution représentant le contenu commun qui doit être disposé dans les zones d'en-tête et de bas de page. Le contenu commun peut être constitué par toute combinaison de contenus caractères, graphique en points et graphique géométrique.

Le document peut comporter un nombre quelconque de contraintes de constitution du type *CommonContent*. *CommonContent* est une classe d'objets logiques composites dont les éléments subordonnés immédiats consistent en une séquence arbitraire d'un ou plusieurs des types de constituants suivants:

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

Quand la structure physique générique est présente, les contraintes de constitution du type *CommonContent* et les contraintes de constitution subordonnées qui leur sont associées doivent obligatoirement être disposées dans des cadres représentant les zones d'en-tête ou de bas de page à l'aide du mécanisme de source logique (voir 6.3.6).

6.2.4.2 CommonText (Texte commun)

CommonText est une contrainte de constitution représentant le contenu commun de caractères qui doit être disposé dans une zone spécifiée de la page.

CommonText est une contrainte de constitution pour une classe d'objets logiques de base qui se rapporte à une ou plusieurs parties de contenu contenant chacune un contenu de caractères retraitable, formaté ou retraitable formaté.

6.2.4.3 PageNumber (Numéro de page)

PageNumber est une contrainte de constitution représentant le contenu commun de caractères qui doit être disposé dans la zone d'en-tête ou de bas de page d'un document. Cette contrainte de constitution est utilisée quand il est nécessaire de présenter un contenu d'en-tête ou de bas de page qui contient un numéro de page produit automatiquement.

PageNumber est une classe d'objets logiques de base qui contient un générateur de contenu. Ce générateur de contenu contient un renvoi à un numéro de page qui est automatiquement évalué quand le document est mis en page. Il est ainsi possible, par exemple, de représenter les numéros de page qui sont affichés sur les pages successives d'un document.

Chaque numéro de page se compose d'un seul nombre qui peut être représenté sous forme de chiffres arabes ou romains ou par son équivalent alphabétique. Les plans de numérotation des pages peuvent commencer à 0 ou à toute valeur supérieure à 0. Le numéro de page produit peut avoir une chaîne de caractères préfixe ou suffixe.

Le format des générateurs de contenu est défini au 6.6.6.8.

6.2.4.4 CommonRaster (Contenu graphique en points commun)

CommonRaster est une contrainte de constitution représentant le contenu graphique en points commun qui doit être disposé dans une zone spécifiée d'une page. Il peut s'agir par exemple d'un logo à insérer dans chaque page du document.

CommonRaster est une classe d'objets logiques de base qui se rapporte à une seule partie de contenu contenant le contenu graphique en points retraitable formaté.

6.2.4.5 CommonGeometric (Contenu graphique géométrique commun)

CommonGeometric est une contrainte applicable au constituant représentant le contenu graphique géométrique qui doit être disposé dans une zone spécifiée d'une page. Il peut s'agir par exemple d'une icône graphique à insérer dans chaque page d'un document.

CommonGeometric est une classe d'objets logiques de base qui se rapporte à une seule partie de contenu contenant un contenu graphique géométrique retraitable formaté.

6.2.4.6 CommonReference (Référence commune)

CommonReference est une contrainte de constitution représentant un contenu caractères commun qui doit être disposé dans une zone définie de la page, et qui représente une chaîne de caractères contenant des références à d'autres parties du document. Cette référence peut consister en une référence à un numéro relatif à un segment, un tableau, une figure, une note de bas de page ou un numéro de page.

CommonReference est une contrainte de constitution pour une classe d'objets logiques de base contenant un générateur de contenu qui, sur interrogation d'évaluation, produit une chaîne de caractères incluant les références. Le format de cette chaîne de référence est défini au 6.6.6.10.

6.2.4.7 CurrentInstance (Application courante)

CurrentInstance est une contrainte de constitution représentant un contenu caractères commun devant être disposé dans une zone spécifiée de la page. Cette contrainte de constitution est spécifiquement utilisée lorsqu'il est nécessaire de se référer à une chaîne de caractères qui est attachée à un constituant logique du document ou à un des constituants physiques Document LayoutRoot, PageSet, RectoPage, VersoPage ou Page. La chaîne numérique peut, par exemple, représenter le titre d'un paragraphe ou d'un tableau qui est contenu ailleurs dans le document.

CurrentInstance est une contrainte de constitution pour une classe d'objets logiques de base contenant un générateur de contenu qui, sur évaluation, produit une copie de la chaîne de caractères associée au document spécifié. Le format de cette référence est défini au 6.6.6.11.

6.2.4.8 TableNumber (Numéro de tableau)

TableNumber est une contrainte de constitution représentant un contenu caractères commun devant être disposé dans une zone spécifiée de la page. Cette contrainte de constitution est spécifiquement utilisée quand il est nécessaire de représenter un numéro de tableau qui doit être placé dans la zone d'en-tête d'un tableau.

TableNumber est une contrainte de constitution pour une classe d'objets logiques de base contenant un générateur de contenu qui, sur évaluation, produit le numéro de tableau requis. Le format du générateur de contenu est défini au 6.6.6.6.

6.2.4.9 CommonNumber (Numéro commun)

CommonNumber est une contrainte de constitution représentant un contenu de caractères commun qui doit être disposé dans une zone spécifiée de la page. Cette contrainte de constitution est spécifiquement utilisée quand il est nécessaire de se référer à un contenu logique consistant en une chaîne numérique qui, par exemple, représente le numéro d'une sous-section ou une page courante.

CommonNumber est une contrainte de constitution pour une classe d'objets logiques de base incorporant un générateur de contenu qui, sur évaluation, génère la référence au numéro requis. Le format de cette référence est défini au 6.6.6.12.

6.3 Caractéristiques physiques

Le présent paragraphe définit les contraintes de constitution de représentation physique fournies par la présente Recommandation pour représenter les caractéristiques d'un document.

On peut utiliser des contraintes de constitution différentes pour représenter et distinguer diverses parties d'un document ayant des caractéristiques physiques différentes. Le présent paragraphe décrit les caractéristiques générales et les utilisations typiques des contraintes de constitution prévues.

Les descriptions des caractéristiques physiques représentées par chacune des contraintes de constitution ne sont fournies qu'à titre indicatif. Il incombe à l'utilisateur de déterminer comment le document devra être représenté au moyen des constituants disponibles. Toutefois, le respect de ces directives permet de faciliter la compréhension du document aux deux extrémités (expéditeur et destinataire).

6.3.1 Vue d'ensemble des caractéristiques physiques

La structure adoptée permet de disposer et présenter un document en un ou plusieurs ensembles de pages. Un ensemble de pages peut être utilisé pour une partie spécifique du document: titre, avant-propos, table des matières, corps du document et appendices.

Chaque ensemble est constitué d'une série de pages. En général, une page peut être subdivisée en trois zones, à savoir le corps, utilisé pour le corps du document proprement dit, et l'en-tête et le bas de page, qui peuvent être utilisés pour le contenu commun.

Il convient de noter que, dans le cas du FOD036 | PM-36, le contenu commun peut aussi être disposé dans le corps du texte ou dans la zone d'en-tête/de note de bas de page.

La présente Recommandation admet trois types de mise en page, spécifiant chacun la position du corps de texte et la présentation du contenu dans le corps du texte. Ces trois types sont appelés mise en page A, B et C et sont définis au 6.3.4.5.

Toutes les applications utilisant la présente Recommandation doivent prendre en charge la mise en page A, et les deux autres types de mise en page pourront être spécifiés à titre facultatif.

La mise en page A est utilisée lorsque le contenu caractères doit être disposé horizontalement (de gauche à droite ou de droite à gauche) et de haut en bas à l'intérieur de la zone de corps. Cette disposition est généralement utilisée pour des documents écrits en caractères latins, hébraïques, arabes ou japonais (dans la plupart des cas).

La mise en page B est utilisée lorsque le contenu caractères doit être disposé verticalement (de bas en haut ou de haut en bas) et de gauche à droite à l'intérieur de la zone de corps. Cette disposition est généralement utilisée pour des documents écrits en caractères latins, hébraïques, arabes ou japonais (dans la plupart des cas) lorsqu'il est nécessaire de présenter le contenu horizontalement (paysage) à l'intérieur de la zone de corps de la page.

La mise en page C est utilisée lorsque le contenu caractères doit être disposé verticalement et de droite à gauche à l'intérieur de la zone de corps. Cette disposition s'applique généralement aux idéogrammes (japonais et chinois par exemple).

La zone de corps, la zone d'en-tête et la zone de bas de page peuvent être encore subdivisées en fonction des impératifs des différentes mises en page. Pour de plus amples détails, se reporter aux 6.3.5 et 6.3.6.

6.3.2 Document LayoutRoot

Document LayoutRoot est une contrainte de constitution qui représente le niveau le plus élevé de la structure physique du document. Ses éléments subordonnés immédiats se composent d'une séquence d'une ou plusieurs contraintes de constitution du type PageSet. Les systèmes de numérotation des pages peuvent être initialisés sur cette contrainte de constitution.

6.3.3 PageSet (Ensemble de pages)

PageSet est une contrainte de constitution qui représente un groupement de pages dans un document. Un ensemble de pages est généralement utilisé pour représenter une partie d'un document qui exige une mise en page spécifique. De même, un ensemble de pages peut correspondre à une partie d'un document qui a une certaine signification logique, par exemple un PageSet peut représenter le premier sujet d'un document ou un chapitre déterminé.

Un seul niveau de PageSet est autorisé dans un document. En revanche, un document peut contenir un nombre quelconque de définitions de classes du type PageSet qui peuvent être utilisées par exemple pour fournir des options de mise en page pour différentes parties d'un document ou pour spécifier les spécifications exactes de mise en page de chaque partie successive d'un document.

Les éléments subordonnés immédiats d'un PageSet consistent en une combinaison de contraintes de constitution des types Page, RectoPage et VersoPage, comme indiqué au 6.3.4.1.

6.3.4 Caractéristiques de page

6.3.4.1 Constituants de page

Trois contraintes de constitution sont prévues pour représenter les pages d'un document, à savoir *Page*, *RectoPage* et *VersoPage*.

Les pages qui forment un ensemble de pages constituent une séquence arbitraire d'une ou plusieurs contraintes de constitution des types précités.

La seule différence dans les caractéristiques de ces types de contraintes de constitution concerne les valeurs qui peuvent être spécifiées pour le paramètre «face» dans l'attribut «type de support». En ce qui concerne la Page, la valeur de ce paramètre peut être spécifiée comme 'recto', 'verso' ou 'non spécifiée'. Pour RectoPage, la valeur de ce paramètre peut être spécifiée comme 'recto' ou 'non spécifiée'; pour VersoPage, la valeur de ce paramètre peut être spécifiée comme

'verso' ou 'non spécifiée'. Les valeurs 'recto' et 'verso' du paramètre «côté de page» de l'attribut «type de support» ne sont pas des valeurs de base.

Les restrictions suivantes s'appliquent en outre aux pages d'un ensemble de pages:

- a) toutes les pages doivent avoir les mêmes dimensions mais peuvent avoir une orientation différente (voir 6.3.4.2);
- b) toutes les pages doivent être formatées sur un support de présentation de même taille (voir 6.3.4.3);
- c) toutes les pages produites à partir d'une classe de pages donnée doivent avoir les mêmes caractéristiques physiques. Autrement dit, une classe de pages donnée ne laisse pas le choix des caractéristiques physiques. Toutefois, les caractéristiques physiques de chaque type de page d'un ensemble de pages peuvent être identiques ou différentes.

Les pages ayant les mêmes caractéristiques physiques sont des pages pour lesquelles la zone de corps, la zone d'en-tête (le cas échéant) et la zone de note de bas de page (le cas échéant) ont les mêmes dimensions et la même position dans la page (voir 6.3.4.3). Toutefois, les pages ayant les mêmes caractéristiques physiques n'ont pas nécessairement la même position sur le support de présentation (voir 6.3.4.4).

6.3.4.2 Dimensions des pages

Les dimensions spécifiées pour les pages peuvent correspondre à n'importe quelle valeur (en BMU) équivalente ou inférieure au format papier A0 de l'ISO ou au format papier E de l'ANSI. Elles peuvent être données en présentation verticale (portrait) ou horizontale (paysage). Les formats japonais B4 et B5 sont également acceptés, mais dans la limite des formats ISO ou ANSI.

Les dimensions équivalentes ou inférieures à la surface de reproduction garantie commune pour les formats A4 de l'ISO et ANSI-A en présentation verticale (portrait) ou horizontale (paysage) sont les valeurs de base. Les plus grands formats de page constituent les formats non essentiels; leur utilisation doit être indiquée dans le profil du document.

Des dimensions de page par défaut peuvent être spécifiées dans le profil du document dans les limites des dimensions maximales définies ci-dessus.

NOTE – Dans un souci d'homogénéité avec les autres références aux dimensions de papier normalisées de l'ANSI, la dimension appelée «North American Letter (NAL)» dans les Rec. de la série T.410 du CCITT | ISO 8613 (par exemple dans la Rec. T.412 du CCITT | ISO 8613-2, article 7) est désignée dans la présente Recommandation par le terme «ANSI-A».

6.3.4.3 Formats de page nominaux

Les formats de page nominaux qui peuvent être spécifiés sont énumérés dans le Tableau 1. Ces formats peuvent être spécifiés en présentation verticale (portrait) ou horizontale (paysage). Toutes les valeurs de format de page nominal sont non essentielles; par conséquent, toutes les valeurs utilisées dans un document doivent être indiquées dans le profil du document.

Tout format de page nominal défini dans le Tableau 1, sous réserve des limitations précisées ci-dessus, peut être spécifié comme la valeur par défaut dans le profil du document.

Le Tableau 1 inclut également la surface de reproduction garantie recommandée (ARA) (*assured reproduction area*). Il peut y avoir perte d'information au stade de la reproduction d'un document si les dimensions associées aux contraintes de constitution de la page type sont supérieures à cette surface pour le format de page nominal spécifié.

6.3.4.4 Décalage de page

Le décalage de page est la distance de la position des bords gauche et supérieur de la page par rapport aux bords gauche et supérieur, respectivement, du support de présentation sur lequel chaque page est reproduite. N'importe quelle valeur de décalage de page peut être spécifiée à condition qu'aucune partie de la surface de la page ne tombe en dehors de la surface de la page nominale. En outre, les décalages de page spécifiés pour les pages initiales, recto et verso d'un ensemble de pages donné peuvent être différents. Le décalage de page par défaut peut être spécifié dans le profil du document.

6.3.4.5 Caractéristiques physiques des pages

6.3.4.5.1 Caractéristiques générales

Chaque page d'un document peut être subdivisée en trois zones rectangulaires, à savoir:

- la zone de corps, qui est réservée au contenu qui constitue le corps du document (voir 6.3.5);
- la zone d'en-tête qui est réservée au contenu de l'en-tête commun (voir 6.3.6);
- la zone de bas de page, qui est réservée au contenu de bas de page commun (voir 6.3.6).

TABLEAU 1/T.506

Tailles des pages nominales

Type de page	Tailles en pouces ou en millimètres	Dimensions en BMU	ARA en BMU
ISO A5	148 mm × 210 mm	7015 × 9920	non défini
ISO A4	210 mm × 297 mm	9920 × 14 030	9240 × 13 200
ISO A3	297 mm × 420 mm	14 030 × 19 840	13 200 × 18 480
ISO A2	420 mm × 594 mm	19 840 × 28 060	18 898 × 27 118
ISO A1	594 mm × 841 mm	28 060 × 39 680	26 173 × 37 843
ISO A0	841 mm × 1189 mm	39 680 × 56 120	37 843 × 54 283
ANSI officiel	8,5" × 14"	10 200 × 16 800	9240 × 15 480
ANSI A	8,5" × 11"	10 200 × 13 200	9240 × 12 400
ANSI B	11" × 17"	13 200 × 20 400	12 744 × 19 656
ANSI C	17" × 22"	20 400 × 26 400	19 500 × 25 800
ANSI D	22" × 34"	26 400 × 40 800	25 800 × 39 600
ANSI E	34" × 44"	40 800 × 52 800	39 600 × 52 200
ANSI F	28" × 40"	33 600 × 48 000	32 400 × 47 400
Japonais officiel	257 mm × 364 mm	12 141 × 17 196	11 200 × 15 300
Japonais lettre	182 mm × 257 mm	8598 × 12 141	7600 × 10 200

La zone de corps est obligatoire; elle doit se trouver sur chaque page d'un document. Les zones d'en-tête et de bas de page sont l'une et l'autre facultatives.

En outre, ces trois zones doivent être entièrement contenues à l'intérieur de la surface de la page et ne doivent pas se chevaucher.

6.3.4.5.2 Body area layout (Mise en page de zone de corps)

Trois types de mise en page de zone de corps sont définis:

- *mise en page A* – Dans ce cas, le sens de formatage dans la zone de corps est spécifié comme étant de 270°;
- *mise en page B* – Dans ce cas, le sens de formatage dans la zone de corps est spécifié comme étant de 0°;
- *mise en page C* – Dans ce cas, le sens de formatage dans la zone de corps est spécifié comme étant de 180°.

6.3.4.5.3 Header/footer area layout (Mise en page de zone d'en-tête/de bas de page)

Quatre types de mise en page de zone d'en-tête/de bas de page sont définis:

- *mise en page de type A1* – Dans ce cas, le sens de formatage dans les zones d'en-tête et de bas de page est spécifié comme étant de 270 degrés. Si la zone d'en-tête ou de bas de page est composite, les sens de formatage dans les cadres de niveau le plus bas sont spécifiés comme étant de 270 degrés;
- *mise en page de type A2* – Dans ce cas, le sens de formatage dans les zones d'en-tête et de bas de page est spécifié comme étant de 0 degré. Si la zone d'en-tête ou de bas de page est composite, le sens de formatage des cadres inférieurs est spécifié comme étant de 270 degrés;

- *mise en page de type B1* – Dans ce cas, le sens de formatage dans les zones d'en-tête et de bas de page est spécifié comme étant de 180 degrés. Si la zone d'en-tête ou de bas de page est composite, les sens de formatage dans les cadres de niveau le plus bas sont spécifiés comme étant de 180 degrés;
- *mise en page de type B2* – Dans ce cas, le sens de formatage dans les zones d'en-tête et de bas de page est spécifié comme étant de 270 degrés. Si la zone d'en-tête ou de bas de page est composite, les sens de formatage dans les cadres de niveau le plus bas sont spécifiés comme étant de 180 degrés.

6.3.4.5.4 Page layout (Mise en page de la page)

Le type de mise en page de la page est déterminé par une combinaison du type de mise en page du corps et du type de mise en page de l'en-tête et du bas de page. Toutes les combinaisons sont autorisées. Toutefois, il est entendu que toutes les applications qui utilisent la présente Recommandation doivent accepter (prendre en charge) les combinaisons mise en page de type A (corps) et mise en page de types A1 et A2 (en-tête et bas de page) alors que la prise en charge d'autres combinaisons peut être facultative.

NOTE – Les combinaisons de mise en page de type A (corps) et mise en page de types A1, B et A1, C et A1 et C et B1 (en-tête et bas de page) sont équivalentes aux mises en page A, B, C et D respectivement dans FOD026 | PM-26.

Dans la mise en page de type A1 ou A2, l'en-tête et le bas de page sont situés en dessus et en dessous de la zone de corps (voir la Figure 1). La Figure 2 illustre la mise en page d'en-tête et de bas de page de types A1 et A2 correspondant à ce cas.

Dans la mise en page de type B1 ou B2, l'en-tête et le bas de page sont situés à droite et à gauche de la zone de corps (voir la Figure 3). La Figure 4 illustre la mise en page d'en-tête et de bas de page de types B1 et B2 correspondant à ce cas.

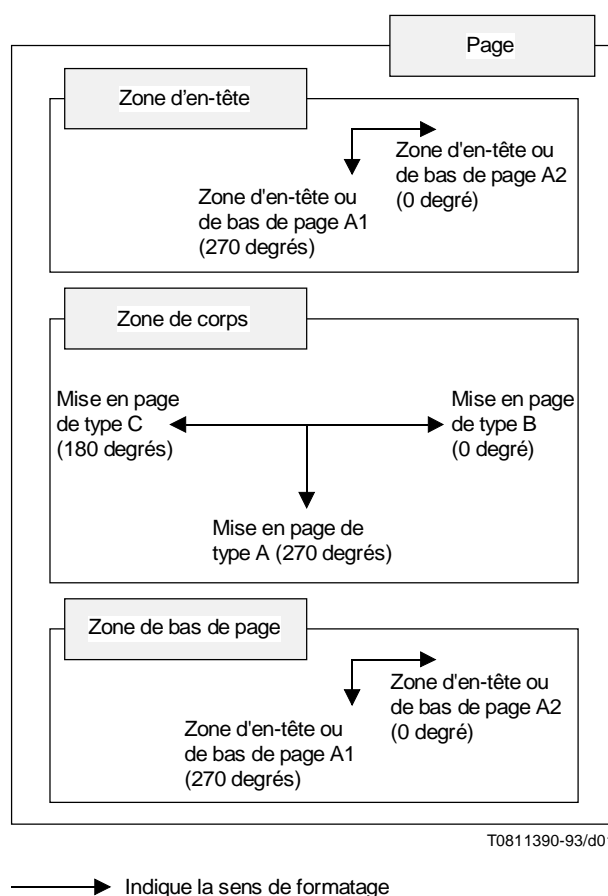


FIGURE 1/T.506

Mise en page de types A, B et C en-tête et bas de page situés respectivement au-dessus et en dessous de la zone de corps

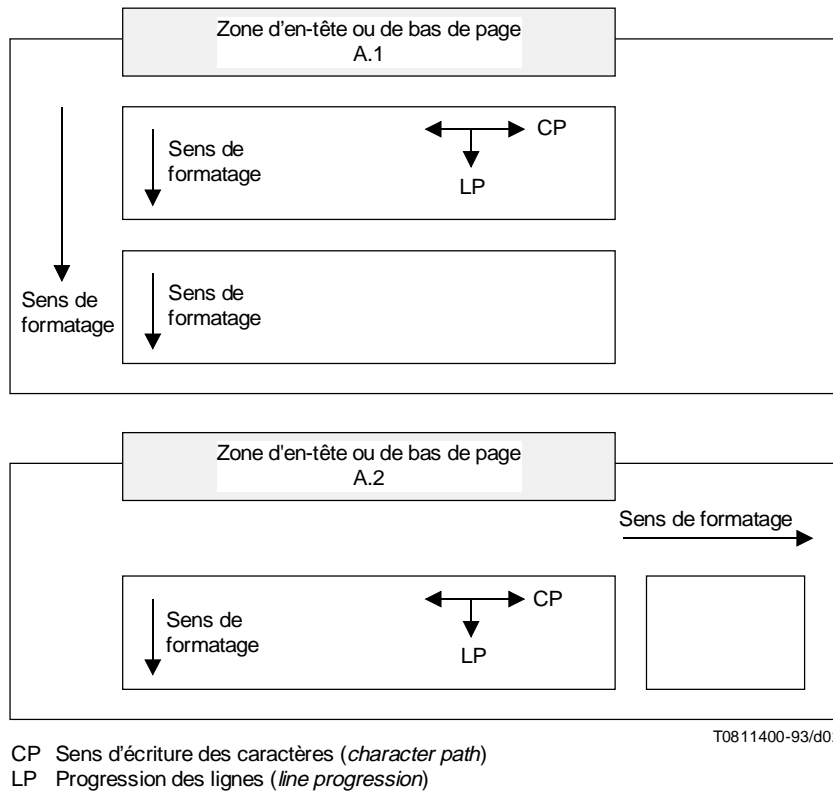


FIGURE 2/T.506

Mise en page de l'en-tête et du bas de page dans le cas illustré à la Figure 1

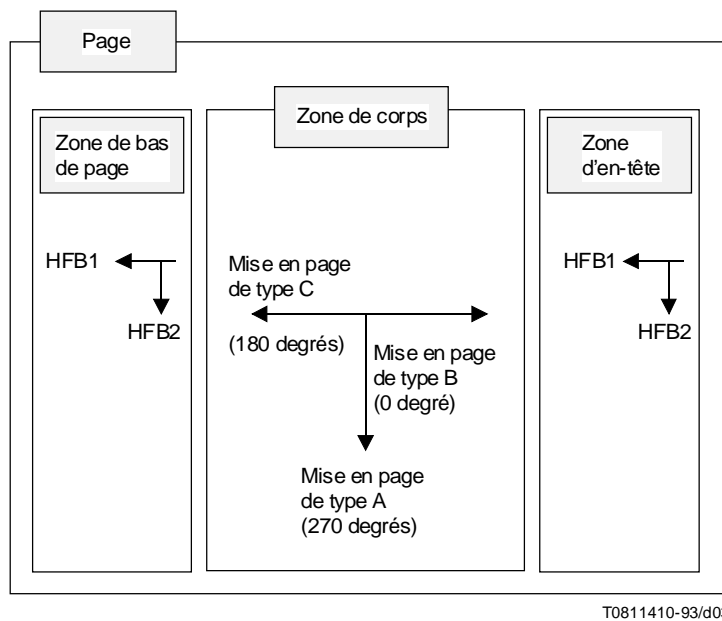
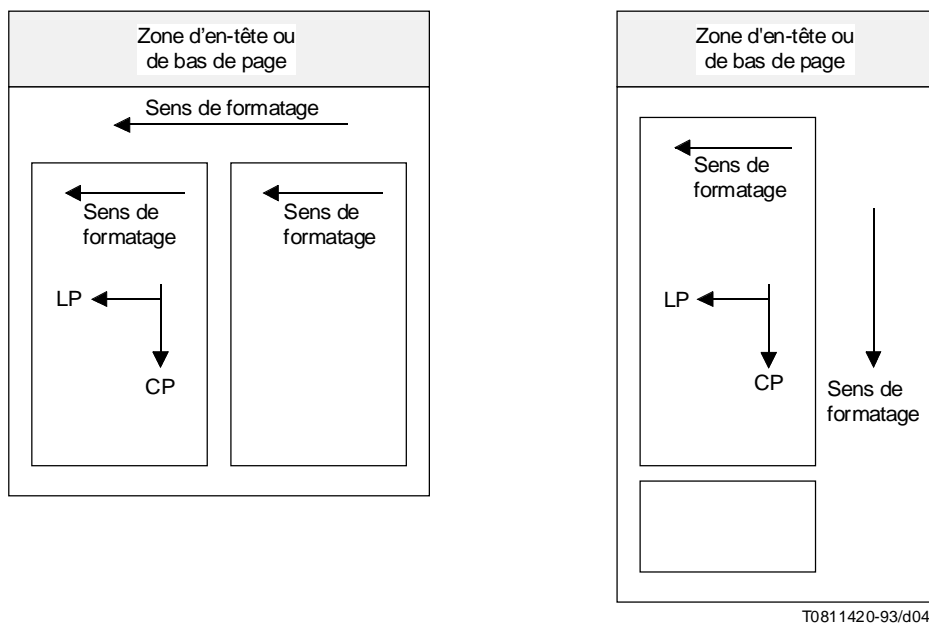


FIGURE 3/T.506

Mise en page de types A, B et C, avec en-tête et bas de page respectivement à droite et à gauche de la zone de corps



CP Sens d'écriture des caractères
LP Progression des lignes

FIGURE 4/T.506

Mise en page des zones d'en-tête et de bas de page
correspondant à la Figure 3

6.3.5 Caractéristiques de la zone de corps

6.3.5.1 Caractéristiques générales

La zone de corps est, dans une page, la zone où est disposé le sujet principal du document, c'est-à-dire corps du document. Le sens de formatage spécifié pour la zone de corps détermine le type de mise en page utilisé.

La zone de corps peut consister en un seul cadre dans lequel le contenu est directement disposé. En ce cas, la zone de corps est représentée par un cadre BasicBody (corps de base).

La zone de corps peut également être subdivisée en différentes régions rectangulaires en fonction des besoins de la mise en page. En ce cas, le cadre du corps est représenté par un cadre VariableCompositeBody ou FixedCompositeBody.

Les zones subordonnées d'un cadre VariableCompositeBody sont représentées par des cadres dont la position et les dimensions sont variables. Ainsi, les zones subordonnées ne sont pas prédéterminées et sont automatiquement ajustées pendant la mise en page en fonction du contenu qui leur est attribué.

Lorsqu'on utilise un cadre FixedCompositeBody, les zones subordonnées sont représentées par des cadres à position fixe, ce qui permet de spécifier la mise en page de la zone de corps avec précision.

Toutefois, si l'on veut disposer à la fois de zones à mise en page fixes et de zones à mise en page variables dans une même zone de corps, il est possible de subordonner un ou plusieurs cadres VariableCompositeBody à un cadre FixedCompositeBody. En ce cas, le sens de formatage pour chacun des cadres VariableCompositeBody peut varier, ce qui permet par exemple de présenter sur la même page des textes rédigés en deux langues différentes.

6.3.5.2 BasicBody (Corps de base)

BasicBody est une contrainte de constitution qui définit le cadre de plus bas niveau représentant une zone de corps dans laquelle le contenu est directement disposé.

La position et les dimensions de ce cadre sont fixes. Le sens de formatage spécifié dépend du type de mise en page utilisé (voir 6.3.4.5).

6.3.5.3 VariableCompositeBody (Corps composite variable)

VariableCompositeBody est une contrainte de constitution qui définit un cadre composite représentant la zone de corps en totalité ou en partie et qui contient un ou plusieurs cadres subordonnés à position variable. La position et les dimensions de ce cadre sont fixes. Le sens de formatage spécifié par ce cadre dépend du type de mise en page utilisé (voir 6.3.1).

Les subordonnés immédiats de ce cadre consistent en une séquence arbitraire d'un ou plusieurs cadres des contraintes de constitution suivantes:

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

Les cadres subordonnés sont tous à positionnement variable et ont des dimensions variables à l'exception des cadres *ArrangedContentVariable*. Ainsi les positions relatives de ces cadres dans la zone de corps peuvent varier et dépendent des positions d'autres cadres (éventuels) placés dans le même cadre *VariableCompositeBody*.

Le sens de formatage des cadres *VariableCompositeBody* peut être spécifié (270, 0 ou 180 degrés). Il détermine le type de mise en page de corps utilisé lorsque *VariableCompositeBody* représente la zone de corps tout entière (voir 6.3.4.5).

Tous les cadres subordonnés immédiats sont disposés dans la direction du sens de formatage spécifié (c'est-à-dire positionnés selon un ordre de remplissage de positionnement «normal»). Les cadres *FootnoteArea* sont disposés dans la direction opposée à celle du sens de formatage (ordre de remplissage de positionnement «inverse»).

Par ailleurs, les cadres doivent avoir le même sens de formatage que le cadre de type *VariableCompositeBody* auquel ils sont subordonnés. Toutefois, cette règle a ses exceptions: les cadres de types *CompositeFixtureVariable*, *CompositeFloat*, *SnakingColumns* et *SourceContentVariable* (se reporter ci-dessous à l'alinéa approprié).

Les Figures 5, 6 et 7 donnent des exemples de présentation de cadres à l'intérieur d'un cadre *VariableCompositeBody* pour les divers types de mise en page.

Un choix de cadres subordonnés énumérés ci-dessus peut être spécifié pour un cadre *VariableCompositeBody*. Différents types de cadre peuvent être choisis au moyen des diverses directives de formatage (voir 6.4), et par conséquent les caractéristiques physiques des zones de corps à l'intérieur d'un ensemble de pages peuvent donc varier d'une page à l'autre à l'intérieur d'un même ensemble de pages.

6.3.5.4 BasicFloat (Colonne unique)

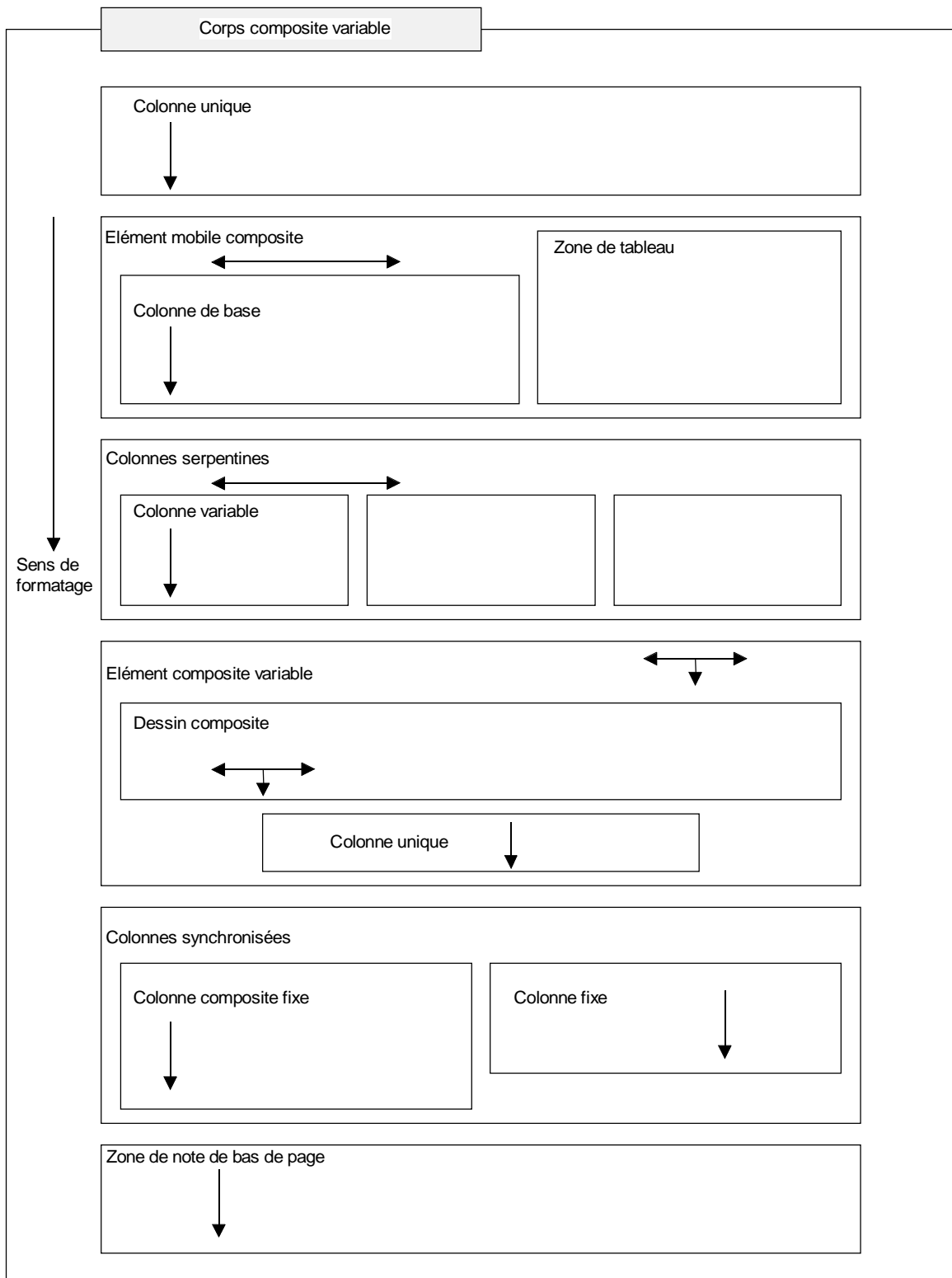
BasicFloat est une contrainte de constitution qui définit le cadre de plus bas niveau servant à représenter une zone de colonnes uniques dans une zone de corps.

Il s'agit d'un cadre à position variable qui peut être spécifié comme étant un subordonné à des cadres de types *VariableCompositeBody*, *CompositeColumnVariable*, *CompositeColumnFixed*, *CompositeFixtureVariable* et *CompositeFixtureFixed*.

La dimension de ce cadre dans la direction perpendiculaire au sens de formatage de la zone de corps a une longueur fixe ou implicitement égale à la valeur maximale autorisée dans la zone de corps.

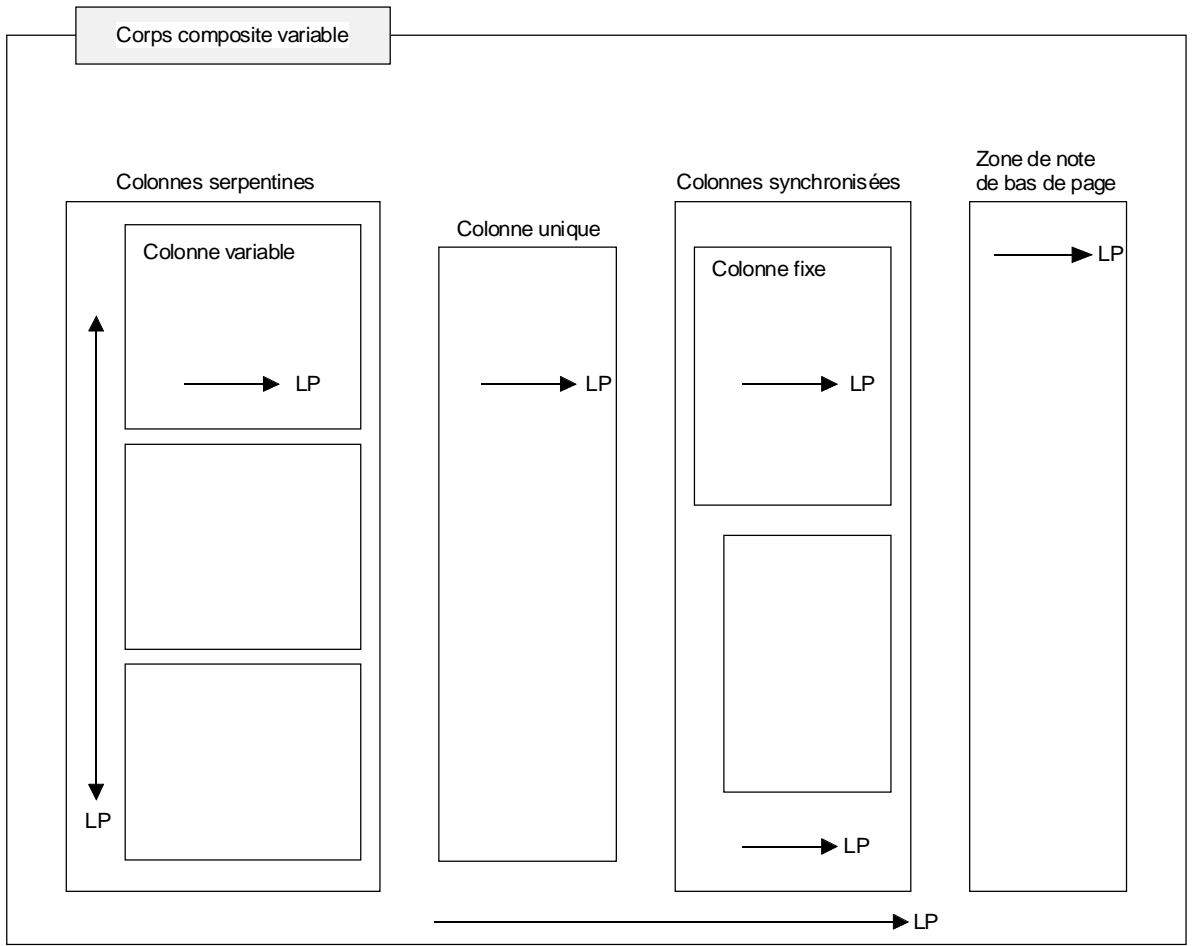
Le côté parallèle à l'orientation de mise en page de la zone de corps a une longueur définie par la «règle B». Cette longueur est donc automatiquement ajustée pendant la mise en page et ramenée à la valeur minimale requise pour présenter tout le contenu destiné au cadre.

Le sens de formatage spécifié pour ce cadre est le même que celui spécifié pour la zone de corps. Dans ce cadre, le contenu doit être présenté selon cette orientation.



T0811430-93/d05

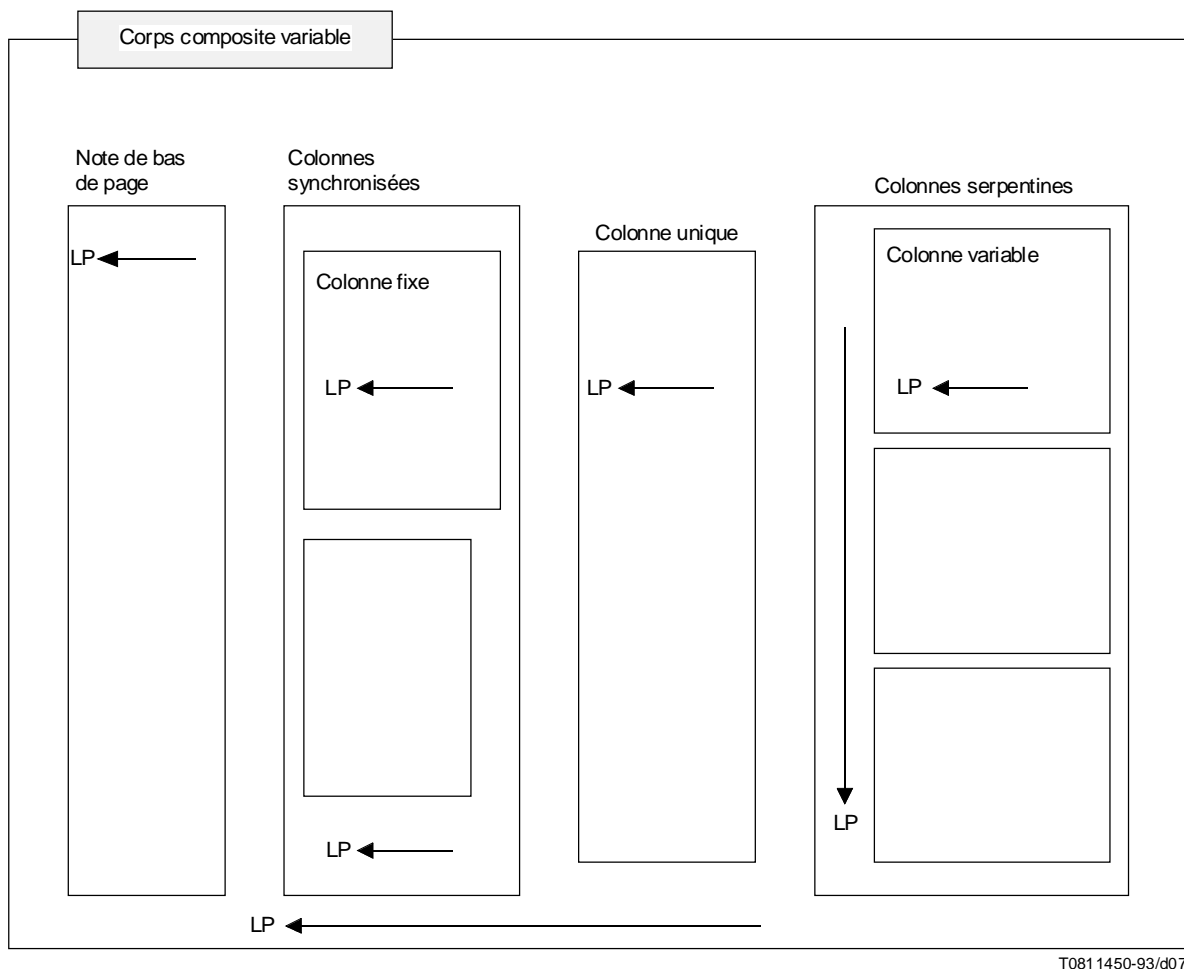
FIGURE 5/T.506
Exemple de formatage de zone de corps – Formatage de type A



LP Sens de formatage (*layout path*)

T0811440-93/d06

FIGURE 6/T.506
Exemple de formatage de zone de corps – Formatage de type B



T081 1450-93/d07

FIGURE 7/T.506

Exemple de formatage de zone de corps – Formatage de type C

6.3.5.5 SnakingColumns (Colonnes serpentes)

SnakingColumns est une contrainte de constitution qui définit un cadre composite représentant une zone de colonnes serpentes dans une zone de corps. Une zone de colonnes serpentes sert généralement à formater une ou plusieurs colonnes dont le contenu peut couler librement d'une colonne à l'autre. Des exemples sont montrés sur les Figures 8 et 9.

Il s'agit d'un cadre à position variable qui ne peut être spécifié que sous forme subordonnée à un cadre de type *VariableCompositeBody*.

Ses subordonnés immédiats consistent en une séquence, ordonnée de façon arbitraire, d'un ou plusieurs cadres des types suivants:

- *ColumnVariable*;
- *CompositeColumnVariable*;
- *ArrangedContentVariable*;
- *SourcedContentVariable*.

La dimension d'un cadre type *SnakingColumns*, dans le sens perpendiculaire au sens de formatage de la zone de corps est fixe ou prend implicitement la valeur maximale autorisée dans la zone de corps.

Parallèlement au sens de formatage de la zone de corps, l'autre dimension de ce cadre est spécifiée par la «règle B». Cette dimension est donc automatiquement ajustée en fonction des cadres subordonnés qu'elle englobe.

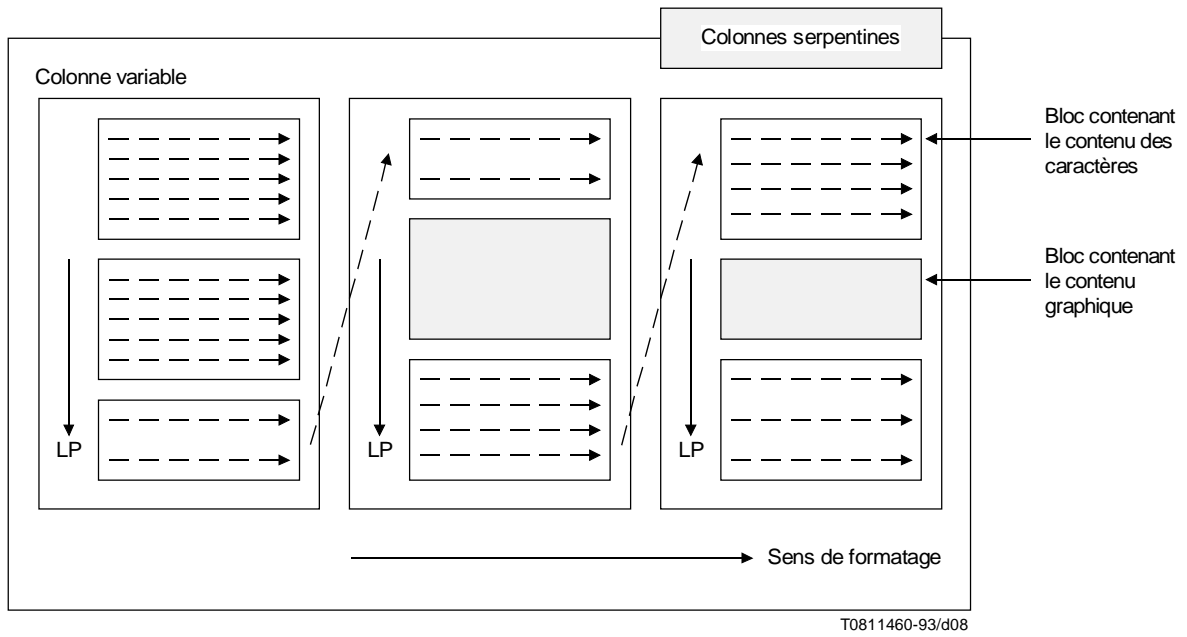


FIGURE 8/T.506
Exemples de formatage d'un cadre de colonnes serpentes

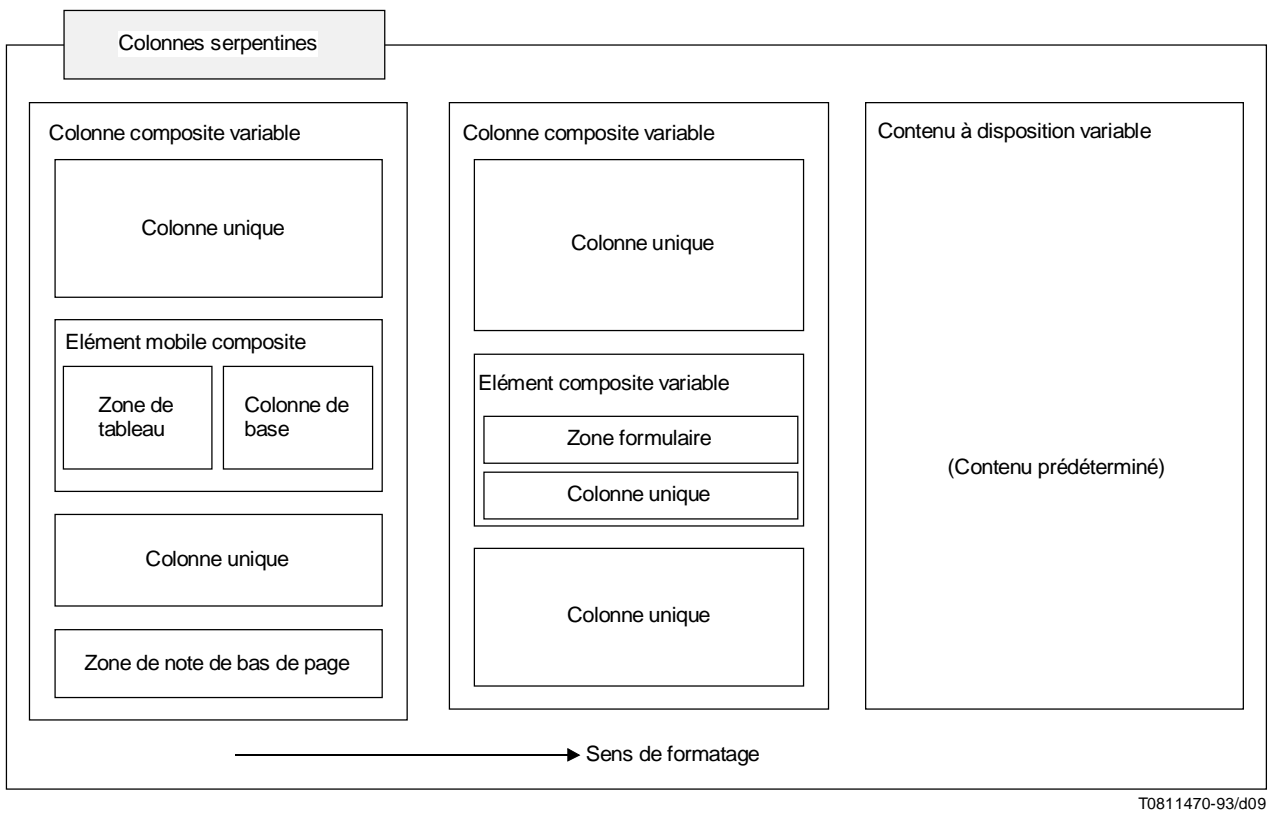


FIGURE 9/T.506
Autre exemple de formatage d'un cadre de colonnes serpentes

Le sens de formatage d'un cadre de type SnakingColumns peut être de 0 ou 180 degrés dans le cas d'une mise en page de type A, de 90 ou 270 degrés dans le cas d'une mise en page de type B et de 270 degrés dans le cas d'une mise en page de type C.

L'attribut «égaliser» peut être spécifié dans le cas d'un cadre de type SnakingColumns pour indiquer que plusieurs cadres subordonnés de type ColumnVariable doivent avoir une longueur approximativement égale dans le sens vertical dans le cas d'une mise en page de type A ou dans le sens horizontal dans le cas des mises en page de types B et C. A noter que «approximativement égale» dans le contexte de l'attribut «égaliser» signifie que les bords avant des objets physiques égalisés sont alignés aussi étroitement que possible sur une ligne perpendiculaire au sens de formatage par ces objets.

6.3.5.6 SynchronizedColumns (Colonnes synchronisées)

Les *SynchronizedColumns* sont une contrainte de constitution qui définit un cadre composite représentant une zone de colonnes synchronisées dans une zone de corps. Une zone de colonnes synchronisées est généralement utilisée pour représenter une ou plusieurs colonnes de contenu tel que le contenu présenté dans chaque colonne appartient à différents flots de formatage. Ainsi, le contenu d'une colonne ne peut pas se couper dans la colonne suivante.

Ce type de disposition de colonnes est en général utilisé lorsqu'il est nécessaire de disposer des contenus distincts parallèlement l'un à l'autre de manière à être alignés. La disposition synchronisée de contenus écrits en langues différentes ou la présentation d'une figure au regard d'un texte en sont deux exemples. La Figure 10 illustre cette configuration.

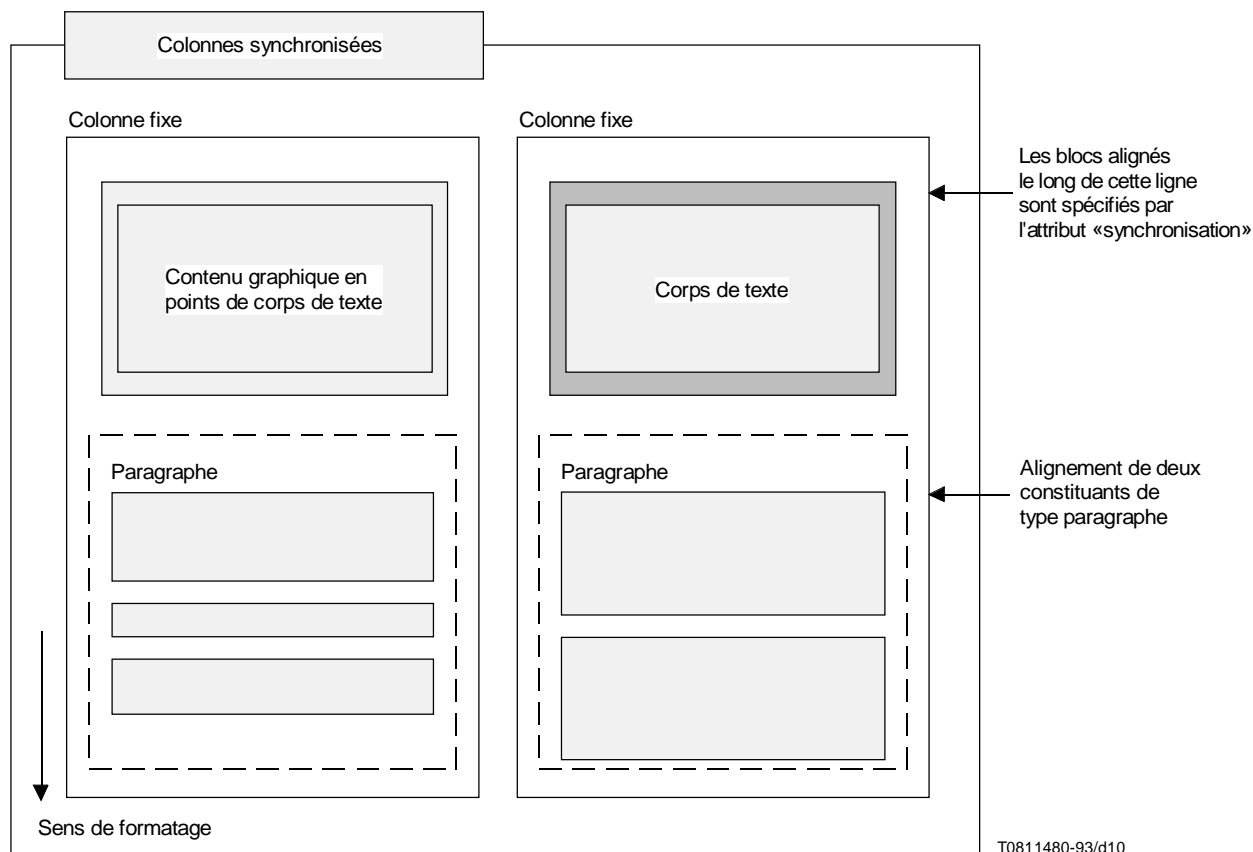


FIGURE 10/T.506
Exemple de disposition de colonnes synchronisées

Les caractéristiques de positionnement et les dimensions des cadres SynchronizedColumns sont les mêmes que celles des cadres SnakingColumns.

Les subordonnés immédiats d'un cadre de type `SynchronizedColumns` consistent en une séquence ordonnée de façon arbitraire d'une ou plusieurs des contraintes de constitution suivantes:

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

Le sens de formatage d'un cadre `SynchronizedColumns` est de 270 degrés pour une mise en page de corps de type A, 0 degré pour une mise en page de corps de type B et 180 degrés pour une mise en page de corps de type C.

6.3.5.7 `ColumnVariable` (Colonne variable)

ColumnVariable est une contrainte de constitution qui définit un cadre du niveau le plus bas servant à représenter une colonne de contenu dans un cadre `SnakingColumns`. Il s'agit d'un cadre à positionnement variable.

La dimension de ce cadre, parallèlement au sens de formatage du cadre supérieur `SnakingColumns` (c'est-à-dire la largeur de la colonne), est fixe. Les dimensions de différents exemples de cadres `ColumnVariable` dans un cadre `SnakingColumns` donné peuvent différer pour permettre de spécifier des colonnes de largeurs différentes.

La dimension dans le sens perpendiculaire au sens de formatage du cadre supérieur (c'est-à-dire la longueur de la colonne), peut être calculée à l'aide de la «règle B» ou spécifiée comme «taille maximale».

Le sens de formatage pour les cadres `ColumnVariable` est de 270 degrés dans le cas de la mise en page A, de 0 degré dans le cas de la mise en page B et de 180 degrés dans le cas de la mise en page C.

Tous les cadres `ColumnVariable` subordonnés au même cadre `SnakingColumns` doivent avoir le même nom de catégorie; des noms différents peuvent être utilisés pour les cadres `ColumnVariable` formatés dans différents cadres `SnakingColumns`.

6.3.5.8 `ColumnFixed` (Colonne fixe)

Columnfixed est une contrainte de constitution qui définit un cadre du niveau le plus bas, servant à représenter une colonne de contenu dans un cadre `FixedCompositeBody` ou `SynchronizedColumns`. Il s'agit d'un cadre à positionnement fixe.

La dimension de ce cadre dans le sens perpendiculaire au sens de formatage du cadre supérieur (c'est-à-dire la largeur de la colonne), peut être fixe ou spécifiée par le sous-paramètre «taille maximale» (voir ci-après). Dans un cadre `SynchronizedColumns` donné, cette dimension peut varier pour des cadres `ColumnFixed` différents, pour permettre de spécifier des colonnes de largeurs différentes. Toutefois, cette largeur doit être spécifiée de sorte que les colonnes ne se chevauchent pas.

La dimension de ce cadre dans la direction parallèle au sens de formatage du cadre supérieur (c'est-à-dire la longueur de la colonne), peut être spécifiée par les sous-paramètres «règle B» ou «taille maximale» s'agissant des mises en page A et B. Dans le cas de la mise en page C, cette dimension peut seulement être spécifiée par le sous-paramètre «taille maximale».

Les cadres `ColumnFixed` subordonnés à un cadre `SynchronizedColumns` donné doivent avoir des noms de catégorie différents.

Le sens de formatage des cadres `ColumnFixed` doit être le même que celui du cadre `SynchronizedColumns` supérieur.

Le contenu présenté dans des cadres `ColumnFixed` différents à l'intérieur du même cadre `SynchronizedColumns` peut être spécifié comme étant «synchronisé» à l'aide de l'attribut «synchronisation».

Les valeurs du sous-paramètre «taille maximale» ne doivent être spécifiées que pour le dernier cadre `ColumnFixed` disposé dans un cadre `SynchronizedColumns` pour empêcher le chevauchement des cadres. En d'autres termes, dans un système coordonné de pages dont le point de référence est dans le coin supérieur gauche, seul le cadre `ColumnFixed` le plus à droite spécifie des valeurs pour le sous-paramètre «taille maximale» en évitant le chevauchement des colonnes.

6.3.5.9 FootnoteArea (Zone note de bas de page)

FootnoteArea est une contrainte de constitution qui définit un cadre du niveau le plus bas utilisé pour représenter une zone réservée à la mise en page des notes de bas de page. Les notes de bas de page peuvent être placées dans des zones du corps ainsi que dans des colonnes et dans des zones réservées à des illustrations à l'intérieur des zones du corps.

Le présent cadre peut être spécifié comme subordonné de cadres des contraintes de constitution suivantes:

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

Il s'agit d'un cadre à positionnement variable, dont l'ordre de positionnement est spécifié comme inverse. Ce type de cadre doit donc être positionné de façon adjacente au bord d'attaque du cadre supérieur.

La dimension des cadres *FootnoteArea*, perpendiculairement au sens de formatage du cadre supérieur, est fixe ou spécifiée par le sous-paramètre «taille maximale». Dans la direction parallèle au sens de formatage, cette dimension est spécifiée par la «règle B», ce qui signifie qu'elle est automatiquement ajustée pour renfermer tout le contenu qui est attribué au cadre.

Le sens de formatage des cadres *FootnoteArea* est le même que celui qui est spécifié pour la zone de corps.

Le seul contenu pouvant être disposé dans ce cadre est le contenu associé aux objets logiques de base subordonnés à l'objet logique composite *FootnoteBody*. A cette fin, l'attribut «catégories permises» de ce cadre spécifie le même nom de catégorie requis pour les objets logiques de base pour les notes de bas de page (voir 6.2.3.10 et 6.2.3.11).

6.3.5.10 FixedCompositeBody (Corps composite fixe)

L'élément *FixedCompositeBody* est une contrainte de constitution qui définit un cadre composite représentant la zone de corps et qui contient un ou plusieurs cadres subordonnés dont la position est fixe. La position et les dimensions de ce cadre sont fixes.

Les subordonnés immédiats des cadres de ce type consistent en une séquence ordonnée de façon arbitraire d'un ou plusieurs cadres des contraintes de constitution suivantes:

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

Les cadres subordonnés peuvent se chevaucher sans restriction.

Le sens de formatage des cadres de type *FixedCompositeBody* peut être de 270, 0 ou 180 degrés. Il détermine le type de mise en page de corps utilisé (voir 6.3.4.5). Il n'a pas d'effet sur le positionnement des cadres mais se répercute sur leurs dimensions puisque certains des cadres énumérés ci-dessus sont de dimensions variables dans une direction donnée.

Par ailleurs, les cadres doivent avoir le même sens de formatage que le cadre de type *FixedCompositeBody* dont ils dépendent. Toutefois, cette règle a des exceptions, à savoir les cadres de types *VariableCompositeBody*, *CompositeFixtureFixed* et *SourcedContentFixed* (voir le paragraphe correspondant ci-après).

Le choix de certains cadres subordonnés appartenant aux types énumérés ci-dessus peut être précisé dans le cas d'un cadre *FixedCompositeBody*. Différents types de cadre peuvent être spécifiés, avec diverses directives de mise en page (voir 6.4), si bien que les caractéristiques de mise en page des zones de corps d'un ensemble de pages donné peuvent varier d'une page à l'autre dans cet ensemble.

6.3.5.11 CompositeFloat (Elément mobile composite)

Un *CompositeFloat* est une contrainte de constitution qui définit un cadre composite spécifiant une zone utilisée pour la mise en page parallèle d'objets complexes – figures, formulaires ou tableaux – et de colonnes de texte simples.

Il s'agit d'un cadre à position variable qui peut être subordonné à un cadre de type *VariableCompositeBody*, *CompositeColumnVariable* ou *CompositeColumnFixed*.

La dimension d'un cadre de type *CompositeFloat*, dans le sens perpendiculaire au sens de formatage de son cadre supérieur, est fixe ou prend implicitement la valeur maximale autorisée.

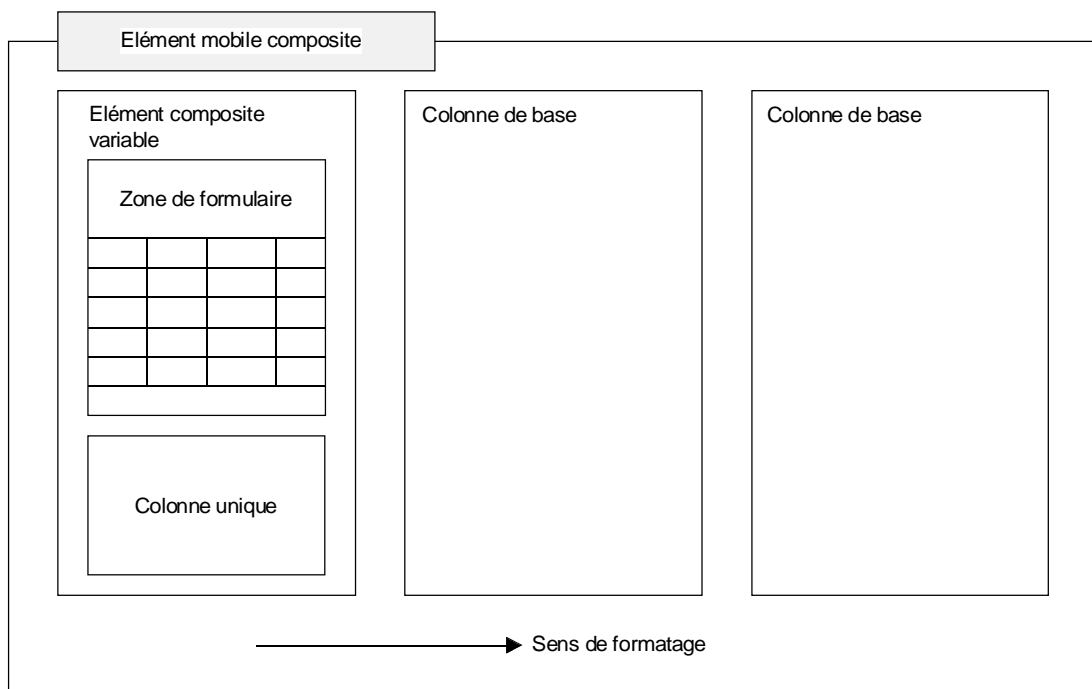
Parallèlement au sens de formatage l'autre dimension est spécifiée par la «règle A». Ainsi, dans ce sens, sa taille est déterminée par la dimension, dans le même sens, du premier cadre mis en page dans un cadre de type *CompositeFloat*.

Les subordonnés immédiats de ce cadre consistent en une séquence, ordonnée de façon arbitraire, d'une ou plusieurs contraintes de constitution suivantes:

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

Le sens de formatage d'un cadre *CompositeFloat* peut être spécifié comme étant de 0 ou 180 degrés dans le cas d'une mise en page de type A ou de 90 ou 270 degrés dans le cas d'une mise en page de type B ou C.

La Figure 11 illustre un exemple type de l'utilisation de ce constituant: forme ou illustration avec contenu de caractères continuant à droite.



T0811490-93/d11

FIGURE 11/T.506

Exemple de cadre de type *CompositeFloat*

6.3.5.12 CompositeFixtureVariable (Elément composite variable)

Un *CompositeFixtureVariable* est une contrainte de constitution qui définit un cadre composite servant à spécifier une zone pour la présentation d'une illustration (figure ou formulaire) à laquelle on associe un texte descriptif et une légende. En conséquence, ce cadre sert essentiellement à présenter des contraintes de constitution logiques du type figure.

Il s'agit d'un cadre à position variable qui peut être subordonné à un cadre *VariableCompositeBody*, *CompositeColumnVariable* ou *CompositeColumnFixed*.

La dimension d'un cadre de type *CompositeFixtureVariable* dans le sens perpendiculaire au sens de formatage du cadre supérieur, est fixe, spécifiée par la «règle B» ou encore prend implicitement la valeur maximale autorisée. Parallèlement au sens de formatage du cadre supérieur, l'autre dimension peut être fixe ou spécifiée par la «règle B».

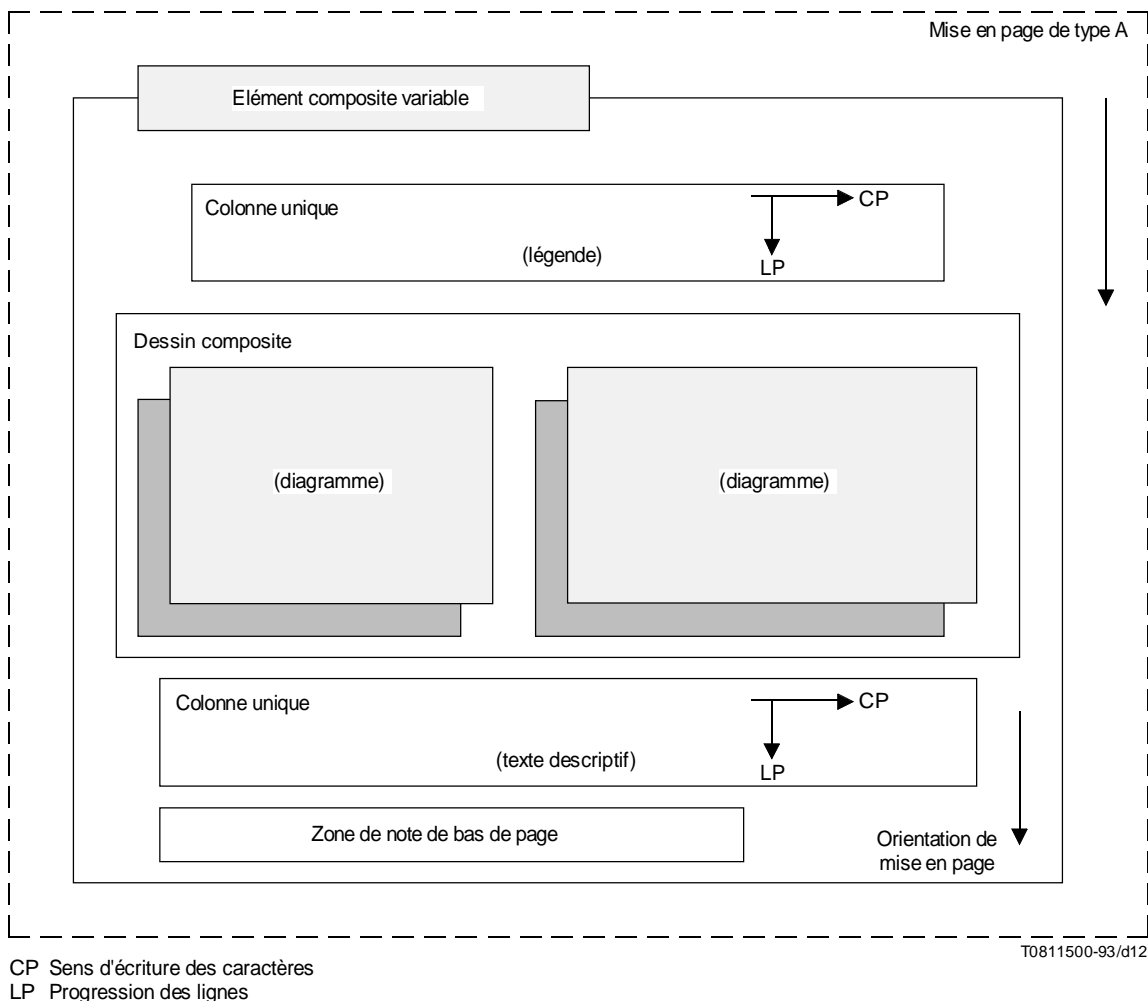
Les subordonnés immédiats de ce cadre consistent en une séquence, ordonnée de façon arbitraire, d'une ou plusieurs des contraintes de constitution suivantes:

- BasicFloat;
- CompositeArtwork;
- FormArea;
- FootnoteArea.

Le sens de formatage d'un cadre *CompositeFixtureVariable* peut être de 0, 180 ou 270 degrés dans le cas d'une mise en page de type A, de 0, 90 ou 270 degrés dans le cas d'une mise en page de type B et de 180 ou 270 degrés dans le cas d'une mise en page de type C.

Tous les cadres subordonnés sont mis en page selon une séquence de remplissage normale, à l'exception des cadres de type *FootnoteArea*, disposés, quant à eux, en ordre inverse.

La Figure 12 illustre la mise en page des cadres de type *CompositeFixtureVariable*.



CP Sens d'écriture des caractères
LP Progression des lignes

FIGURE 12/T.506

Exemple de cadre de type *CompositeFixtureVariable*

6.3.5.13 CompositeFixtureFixed (Elément composite fixe)

Un *CompositeFixtureFixed* est une contrainte de constitution qui définit un cadre composite ayant les mêmes caractéristiques qu'un cadre *CompositeFixtureVariable*, position et dimensions exceptées.

Ce cadre peut être subordonné à un cadre de type *FixedCompositeBody*.

6.3.5.14 CompositeArtwork (Dessin composite)

CompositeArtwork est une contrainte de constitution définissant un cadre composite représentant une zone qui contient une illustration de type figure ou diagramme. Cette contrainte sert à mettre en page les constituants logiques du type *Artwork* qui sont subordonnés aux contraintes de constitution logiques du type figure.

Il s'agit d'un cadre à positionnement variable pouvant être spécifié comme subordonné d'un cadre *CompositeFixtureVariable* ou *CompositeFixtureFixed*.

Les dimensions de ce cadre (perpendiculairement et parallèlement au sens de formatage du cadre supérieur) peuvent être fixées séparément ou spécifiées par le sous-paramètre «règle B».

Les subordonnés immédiats d'un cadre *CompositeArtwork* consistent en une séquence d'un ou plusieurs cadres de niveau le plus bas du type *BasicFixture* renfermant un contenu caractères, un contenu graphique en points ou un contenu graphique géométrique. Le chevauchement de cadres *BasicFixture* permet de former des images composites.

Le sens de formatage d'un cadre *CompositeArtwork* doit être le même que celui du cadre supérieur. Compte tenu de cette restriction, le sens de formatage d'un cadre *CompositeArtwork* peut donc être de 0, 180 ou 270 degrés dans le cas de la mise en page A, de 0, 90 ou 270 degrés dans le cas de la mise en page B et de 180 ou 270 degrés dans le cas de la mise en page C.

6.3.5.15 BasicFixture (Elément de base)

BasicFixture est une contrainte de constitution qui définit un cadre du niveau le plus bas spécifiant une zone de mise en page du contenu à l'intérieur d'un *FixedCompositeBody* et d'un *CompositeArtwork*.

Le positionnement de ce cadre est fixe à l'intérieur du cadre supérieur. Ses dimensions (perpendiculairement et parallèlement au sens de formatage du cadre supérieur) peuvent être fixées séparément ou spécifiées à l'aide de la «règle B».

Si le cadre supérieur d'un cadre *BasicFixture* est *FixedCompositeBody*, le sens de formatage du cadre *BasicFixture* est positionné pour être le même que celui de son cadre supérieur.

Si le cadre supérieur d'un cadre *BasicFixture* est *CompositeArtwork*, le sens de formatage du cadre *BasicFixture* est limité comme suit:

- si le sens de formatage du cadre supérieur est fixé à 270°, le sens de formatage de *BasicFixture* peut être fixé à 270° ou 180°;
- si le sens de formatage du cadre supérieur est fixé à 0°, le sens de formatage de *BasicFixture* peut être fixé à 0° ou 270°;
- si le sens de formatage du cadre supérieur est fixé à 180°, le sens de formatage de *BasicFixture* peut être fixé à 180° ou 270°.

6.3.5.16 FormArea (Zone formulaire)

FormArea est une contrainte de constitution qui définit un cadre composite représentant une zone à dimensions fixes réservée à la mise en page d'une illustration de type formulaire.

Cette zone est subdivisée en une ou plusieurs zones simples et composites. On entend par zone simple une zone dans laquelle le contenu est directement présenté, et par zone composite une zone qui se divise en une ou plusieurs zones simples et composites.

Il s'agit d'un cadre à positionnement variable pouvant être spécifié comme subordonné d'un cadre *CompositeFixtureVariable* ou *CompositeFixtureFixed*.

Les dimensions de ce cadre sont fixes dans les deux directions.

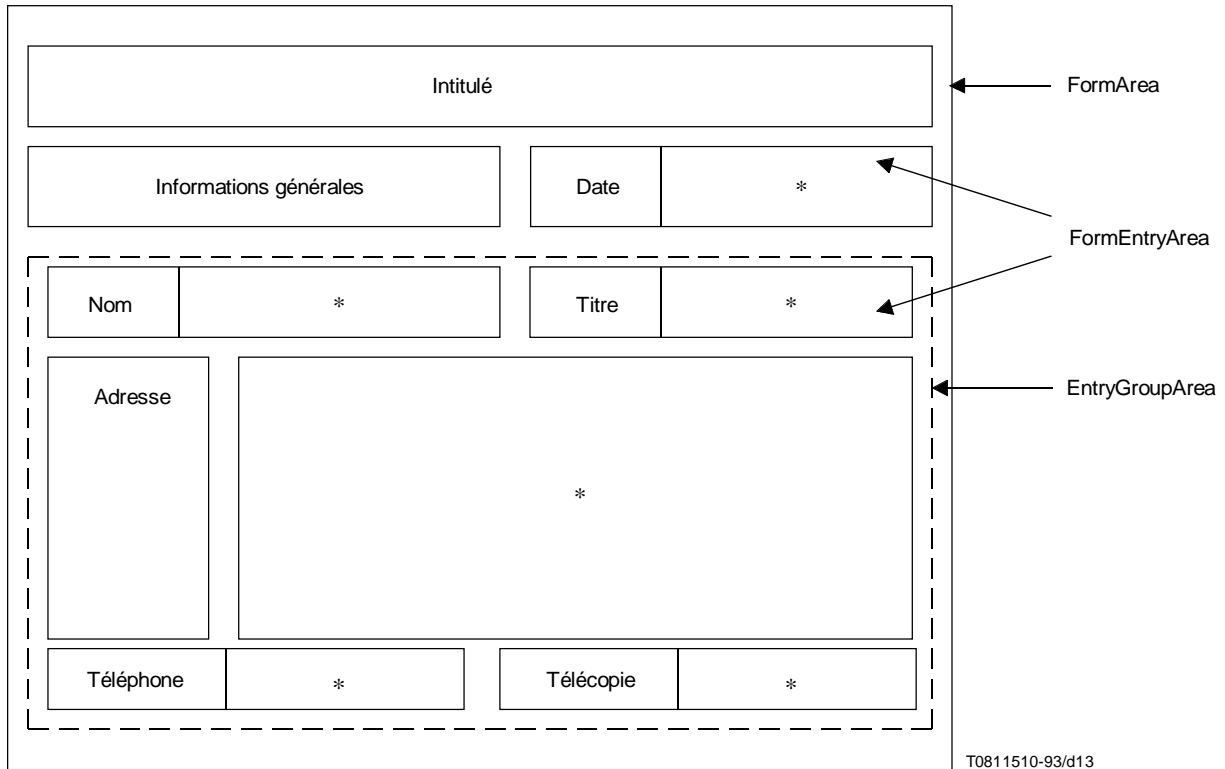
Les subordonnés immédiats d'un cadre *FormArea* consistent en une séquence, ordonnée de façon arbitraire, d'un ou plusieurs cadres à positionnement fixe des contraintes de constitution suivantes:

- *FormEntryArea*;
- *EntryGroupArea*;
- *ArrangedContentFixed*.

Dans un formulaire, les cadres de types `FormEntryArea` et `ArrangedContentFixed` représentent les zones simples, les cadres `EntryGroupArea` représentent les zones composites.

Le sens de formatage d'un cadre `FormArea` est implicitement de 270 degrés et n'influe pas sur le sens de formatage des cadres subordonnés.

La Figure 13 présente un exemple de formatage d'un formulaire.



* Cases remplies par l'utilisateur

FIGURE 13/T.506
Exemple de mise en page d'un formulaire

6.3.5.17 `EntryGroupArea` (Zone de groupe d'entrée)

`EntryGroupArea` est une contrainte de constitution qui représente une zone composite à l'intérieur d'un cadre `FormArea`. Les positions et dimensions de ce cadre sont fixes dans les deux directions. La valeur par défaut du sens de formatage de ce cadre est 270°.

Les subordonnés immédiats de ce cadre consistent en une séquence ordonnée de façon arbitraire d'un ou plusieurs cadres à positionnement fixe du type `FormEntryArea`.

6.3.5.18 `FormEntryArea` (Zone entrée formulaire)

`FormEntryArea` est une contrainte de constitution qui définit un cadre du niveau le plus bas spécifiant une zone de mise en page du contenu à l'intérieur d'un cadre `FormArea`. Ce cadre peut contenir les éléments contenu caractères, contenu graphique en points ou contenu graphique géométrique.

La position et les dimensions de ce cadre sont fixes. Son sens de formatage est de 270 degrés, quel que soit le type de mise en page.

6.3.5.19 BasicColumn (Colonne de base)

BasicColumn est une contrainte de constitution qui définit un cadre du niveau le plus bas spécifiant une zone de mise en page d'un contenu de type colonne à l'intérieur d'un cadre CompositeFloat. Le positionnement de ce cadre est variable à l'intérieur du cadre supérieur.

Perpendiculairement à l'orientation de mise en page du cadre supérieur, la dimension de ce cadre peut être fixe ou spécifiée par le sous-paramètre «règle B». Parallèlement à cette orientation, son autre dimension peut être fixe, spécifiée par le sous-paramètre «règle B» ou prendre implicitement la valeur taille maximale. Sa taille peut donc être ajustée automatiquement en fonction du contenu qui lui est attribué.

Le sens de formatage de ce cadre est de 270 degrés dans le cas de la mise en page A, de 0 degré dans le cas de la mise en page B et de 180 degrés dans le cas de la mise en page C.

6.3.5.20 CompositeColumnVariable (Colonne composite variable)

CompositeColumnVariable est une contrainte de constitution qui définit un cadre composite spécifiant une zone représentant une colonne à l'intérieur d'un cadre SnakingColumns. Cette colonne est subdivisée en plusieurs zones permettant de répondre à différents besoins de mise en page. Par exemple, on peut utiliser le présent cadre pour représenter une colonne contenant un tableau et un diagramme complexe intégré dans un train de données de type contenu caractères. Il s'agit d'un cadre à positionnement variable.

Parallèlement au sens de formatage du cadre SnakingColumns supérieur, la dimension de ce cadre (c'est-à-dire la largeur de la colonne), est fixe. Les dimensions de différents cadres CompositeColumnVariable à l'intérieur d'un même cadre SnakingColumns peuvent varier, ce qui permet de spécifier des colonnes de largeurs différentes.

Perpendiculairement au sens de formatage du cadre supérieur, l'autre dimension de ce cadre (c'est-à-dire la longueur de la colonne), peut être calculée à l'aide du sous-paramètre «règle B» ou spécifiée comme «taille maximale».

Les subordonnés immédiats de ce cadre se composent d'une séquence, ordonnée de façon arbitraire, de cadres des contraintes de constitution suivantes:

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

Le sens de formatage des cadres ColumnVariable est de 270 degrés dans le cas de la mise en page A, de 0 degré dans le cas de la mise en page B et de 180 degrés dans le cas de la mise en page C.

Tous les cadres subordonnés sont mis en page dans l'ordre de remplissage normal, à l'exception de FootnoteArea qui est présenté dans l'ordre inverse.

6.3.5.21 CompositeColumnFixed (Colonne composite fixe)

CompositeColumnFixed est une contrainte de constitution qui définit un cadre composite spécifiant une zone représentant une colonne à l'intérieur d'un cadre SynchronizedColumns. La colonne est subdivisée en plusieurs zones permettant de répondre à différents besoins de mise en page.

Ce cadre possède les mêmes caractéristiques que les cadres CompositeColumnVariable, à l'exception de sa position qui est fixe. Ses dimensions sont les mêmes que celles des cadres ColumnFixed.

6.3.5.22 CompositeCommon (Composite commun)

CompositeCommon est une contrainte de constitution qui définit un cadre composite spécifiant à l'intérieur de la zone de corps une zone devant renfermer un contenu commun. Cette zone peut être subdivisée, ce qui permet de mettre en page différents types de contenu commun.

Ce cadre a un positionnement et des dimensions fixes. Il peut être spécifié comme subordonné d'un cadre FixedCompositeBody.

Les subordonnés d'un cadre CompositeCommon peuvent être composés:

- a) d'un nombre et d'une combinaison quelconques de cadres à positionnement variable des types SourcedContentVariable et ArrangedContentVariable; ou
- b) d'un nombre et d'une combinaison quelconques de cadres à positionnement fixe des types SourcedContentFixed et ArrangedContentFixed.

Dans le cas b) ci-dessus, les cadres subordonnés peuvent se chevaucher sans restriction.

Le sens de formatage d'un cadre CompositeCommon est respectivement de 270, 0 et 180 degrés dans les cas des mises en page A, B et C.

6.3.5.23 Constituants utilisés pour la présentation des tableaux

Le présent paragraphe vise à définir les contraintes applicables à la présentation des tableaux. Le paragraphe 6.3.5.23.1 donne une vue d'ensemble des caractéristiques de présentation relatives aux tableaux et les paragraphes suivants définissent les différents constituants.

6.3.5.23.1 Vue d'ensemble

Un tableau se compose de trois zones principales:

- une zone d'en-tête unique, facultative, placée en haut du tableau;
- une zone unique d'étiquette, facultative, placée immédiatement en dessous de la zone d'en-tête;
- une ou plusieurs zones de ligne, placée(s) consécutivement en dessous de la zone d'en-tête et de la zone d'étiquette du tableau.

La zone d'en-tête du tableau sert généralement à contenir un titre ou une légende décrivant l'objet du tableau. Cette zone comporte une ou plusieurs subdivisions, chacune d'entre elles contenant le contenu commun extrait de la structure logique à l'aide du mécanisme source logique.

La zone d'étiquette du tableau sert généralement à contenir les étiquettes correspondant aux colonnes du tableau. Les zones de ligne forment le corps principal du tableau et servent à présenter les informations faisant l'objet du tableau. Ces zones peuvent être simples ou composites, comme il est indiqué ci-dessous.

La Figure 14 présente un exemple de tableau simple dans lequel l'étiquette du tableau et chaque ligne sont composées d'une séquence de zones appelées cellules disposées horizontalement dans la zone de tableau. Ces cellules constituent des exemples de zones simples d'étiquette et de ligne de tableau.

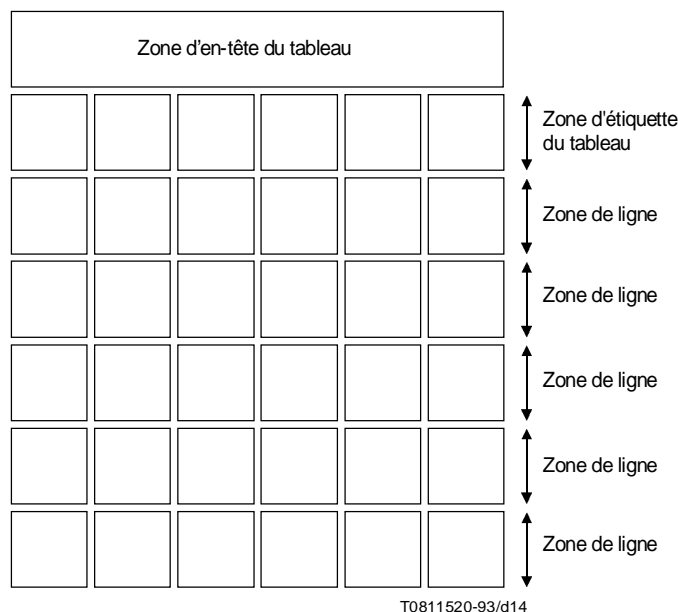


FIGURE 14/T.506

Exemple de tableau *simple*

La Figure 15 présente un exemple de tableau plus complexe illustrant l'utilisation des zones composites d'étiquette et de ligne de tableau. Une zone composite d'étiquette de tableau se compose d'une séquence d'au moins deux sous-lignes, chacune d'entre elles étant divisée en cellules séparées. Selon que les étiquettes se rapportent à une seule ou à plusieurs colonnes, les cellules de chaque ligne peuvent être ou non de la même taille. La séquence de sous-lignes peut être précédée, sur la gauche, d'une cellule unique qui peut être utilisée, par exemple, pour contenir le titre du groupe de sous-lignes.

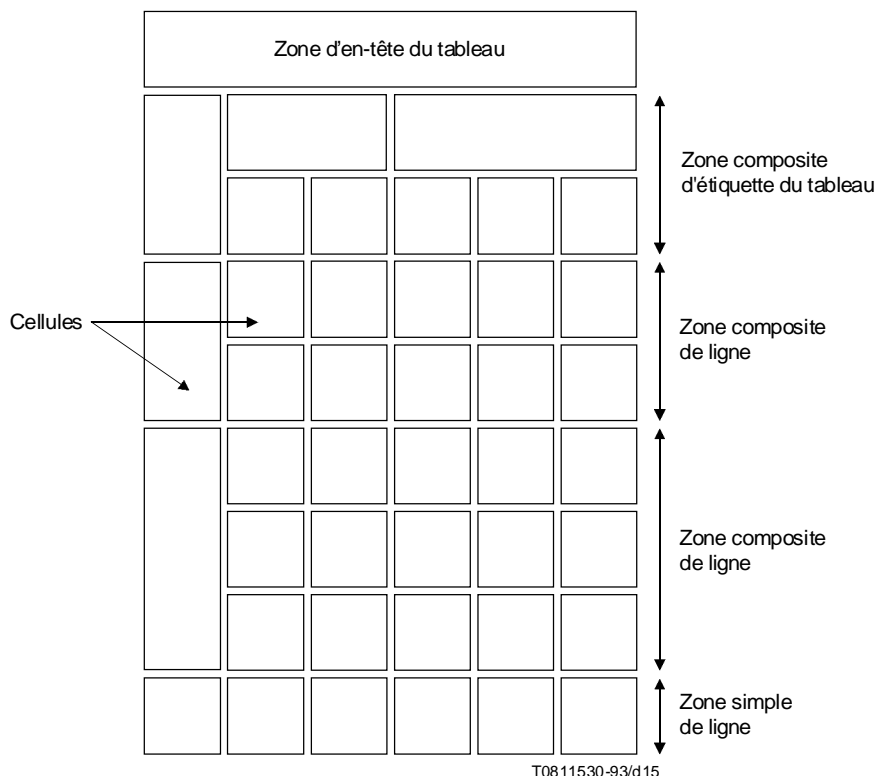


FIGURE 15/T.506
Exemple de tableau *complexe*

La structure générale des zones composites de ligne est la même que celle des zones composites d'étiquette de tableau. On obtient une gamme étendue de types de tableaux en associant différentes combinaisons de zones simples et composites d'étiquette et de ligne de tableau.

Les cadres utilisés pour les zones d'étiquette et de ligne de tableau sont respectivement représentés par les Figures 16 et 17. Les zones appelées cellules sont destinées à recevoir un contenu d'un seul type. Les cadres qui représentent ces cellules ont une position et une dimension horizontale fixes. Cependant, la dimension verticale d'une cellule peut être spécifiée comme variable de façon à être automatiquement ajustée au cours du processus de mise en page pour permettre à la cellule de renfermer tout le contenu qui lui est attribué.

Dans le cas d'une étiquette ou d'une ligne de tableau composite, la dimension verticale du cadre contenant (c'est-à-dire des cadres `LabelComponent` ou `SubRow`, respectivement) est également variable: elle peut ainsi être ajustée automatiquement au cours du processus de mise en page pour que ce cadre soit suffisamment large pour contenir la cellule la plus large de la ligne concernée.

De même, tous les cadres contenant (`CompositeTableLabel`, `TableLabel`, `SubRowGroup` et `SubRow`) peuvent être spécifiés comme ayant une dimension verticale variable de façon que la dimension verticale de chaque zone d'étiquette et de ligne du tableau puisse être ajustée automatiquement pour prendre en compte toutes les informations devant être présentées dans ces zones.

Le cadre qui sert à spécifier la zone de tableau complète (de type cadre `TableArea`, non représenté sur les figures) peut aussi avoir une dimension verticale variable.

Le mécanisme d'attribution du contenu aux cellules d'un tableau est décrit au 6.4.1.3.8.

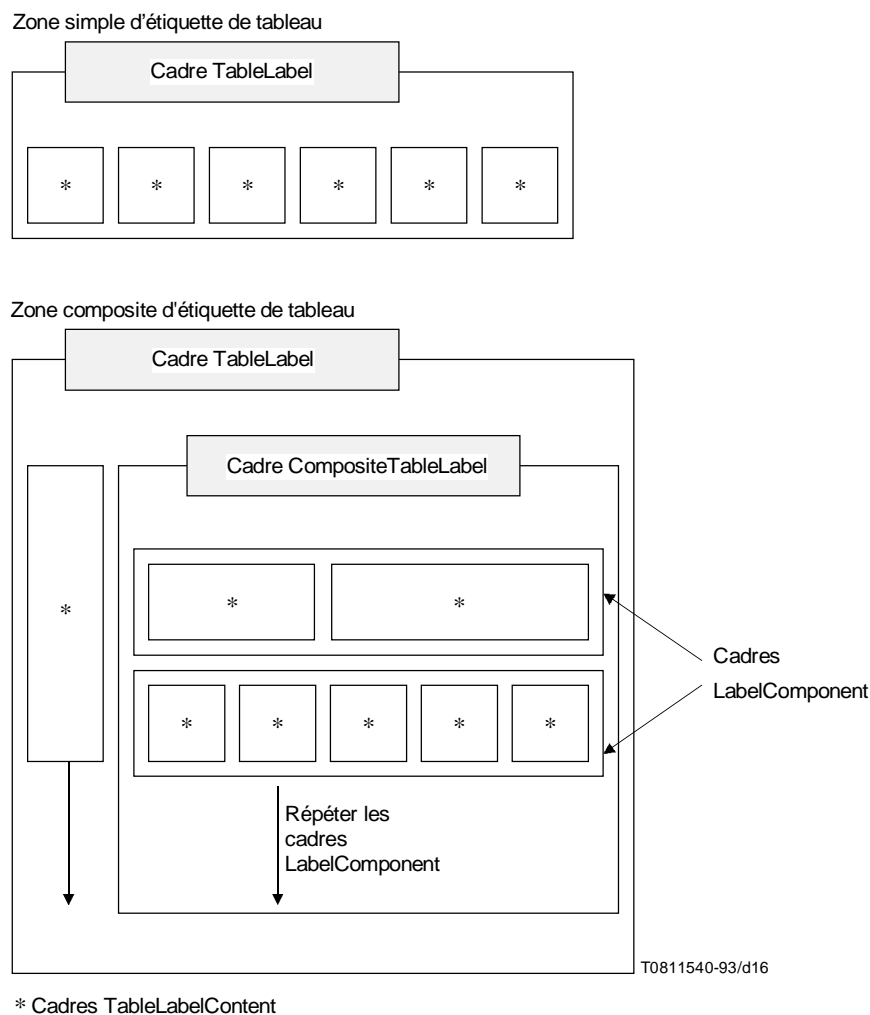


FIGURE 16/T.506
Cadres utilisés pour la représentation des étiquettes de tableau

6.3.5.23.2 TableArea (Zone tableau)

TableArea est une contrainte de constitution, qui définit un cadre composite utilisé pour spécifier une zone réservée à la présentation d'un tableau. Les subordonnés de cette contrainte de constitution peuvent être les suivants:

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

Il s'agit d'un cadre à positionnement variable. Perpendiculairement au sens de formatage du cadre supérieur, sa dimension est fixe. Parallèlement à cette orientation, son autre dimension est fixe ou spécifiée par la «règle B». Son orientation de mise en page est de 270 degrés.

Les subordonnés immédiats de cette contrainte de constitution sont *TableHeader* (facultatif), suivi de *TableLabel* (facultatif), suivi d'une séquence d'une ou plusieurs contraintes de constitution du type *RowArea* et d'un *TableLabel* (facultatif).

6.3.5.23.3 TableHeader (En-tête tableau)

TableHeader est une contrainte de constitution, qui spécifie un cadre composite délimitant une zone à l'intérieur d'un cadre *TableArea*. Cette contrainte sert généralement à présenter les informations d'en-tête relatives à un tableau.

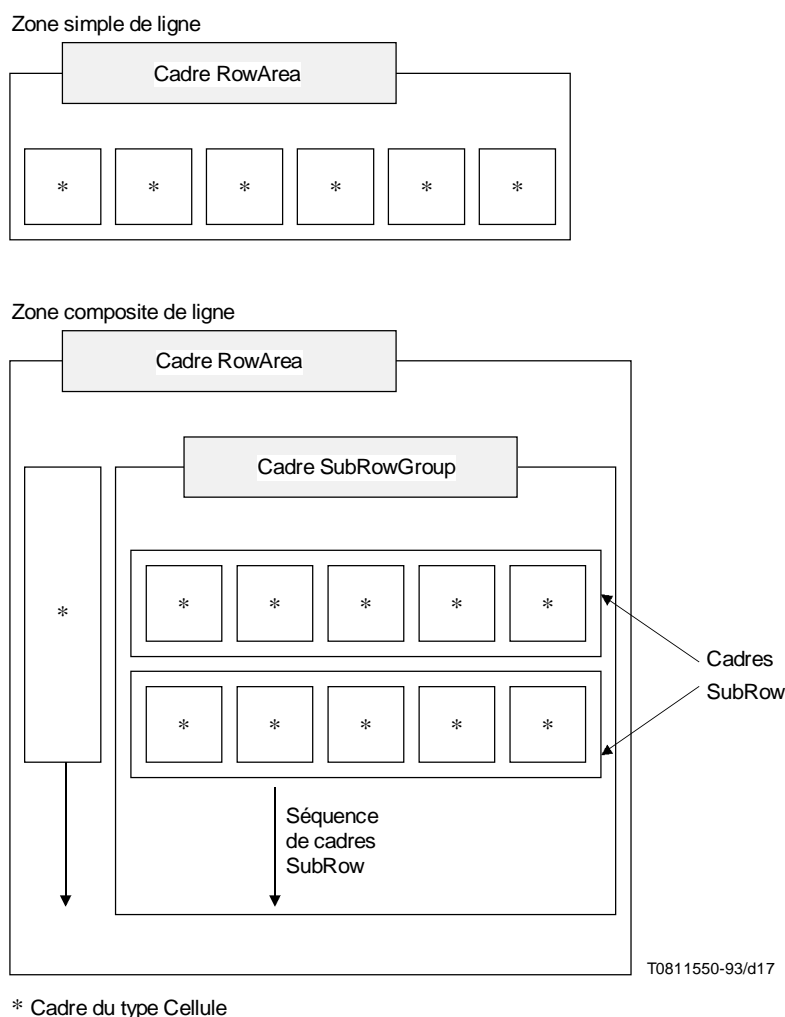


FIGURE 17/T.506
Cadres utilisés pour la représentation des lignes de tableau

Il s'agit d'un cadre à positionnement variable. La dimension de ce cadre, perpendiculairement au sens de formatage du cadre supérieur, est fixe. Parallèle à ce sens, son autre dimension est fixe ou spécifiée par le sous-paramètre «règle B».

Les subordonnés immédiats de ce constituant sont composés d'une séquence de constituants du type SourcedContentFixed. Le contenu présenté dans un cadre TableHeader est donc extrait des contraintes de constitution logiques du type CommonContent.

6.3.5.23.4 TableLabel (Étiquette tableau)

TableLabel est une contrainte de constitution, qui définit un cadre composite spécifiant une zone à l'intérieur d'un cadre TableArea. Cette contrainte sert à la présentation des étiquettes des colonnes d'informations du tableau.

Il s'agit d'un cadre à positionnement variable dont la dimension, perpendiculaire au sens de formatage du cadre supérieur, est fixe ou spécifiée par le sous-paramètre «règle-B».

Les subordonnés immédiats de cette contrainte de constitution sont composés:

- a) d'une séquence d'une ou plusieurs contraintes de constitution du type TableLabelContent; ou
- b) d'une séquence composée d'une contrainte de constitution du type TableLabelContent et d'une contrainte de constitution du type CompositeTableLabel.

6.3.5.23.5 CompositeTableLabel (Étiquette composite tableau)

CompositeTableLabel est une contrainte de constitution, qui définit un cadre composite spécifiant une zone comportant un cadre *TableLabel* pour le formatage d'une étiquette composite de tableau.

Il s'agit d'un cadre à positionnement fixe dont la dimension, perpendiculaire au sens de formatage du cadre supérieur, est fixe, et dont l'autre dimension (parallèle à ce sens), peut être fixe, spécifiée par le sous-paramètre «règle B» ou implicitement égale à la taille maximale autorisée.

Les subordonnés immédiats de cette contrainte de constitution se composent d'une séquence d'une ou plusieurs contraintes de constitution du type *LabelComponent*.

6.3.5.23.6 LabelComponent (Composant étiquette)

LabelComponent est une contrainte de constitution, qui définit un cadre composite spécifiant une zone à l'intérieur d'un cadre *CompositeTableLabel*. Cette contrainte sert à présenter une ligne d'étiquettes à l'intérieur d'un en-tête de tableau.

Il s'agit d'un cadre à position variable. Sa dimension perpendiculaire au sens de formatage du cadre supérieur est fixe. Sa dimension parallèle au sens de formatage est également fixe, telle que spécifiée par le sous-paramètre «règle B» ou de valeur par défaut, la taille maximale autorisée.

Les subordonnés immédiats de ce cadre sont composés d'une séquence de contraintes de constitution du type *TableLabelContent*.

6.3.5.23.7 TableLabelContent (Contenu étiquette tableau)

TableLabelContent est une contrainte de constitution, qui définit un cadre du niveau le plus bas délimitant une zone à l'intérieur d'un cadre *TableLabel* ou *LabelComponent*. Cette contrainte est utilisée pour la présentation d'informations d'en-tête relatives à une ou plusieurs colonnes d'un tableau. Les éléments contenu caractères, contenu graphique en points ou contenu graphique géométrique peuvent être attribués à ce cadre.

Il s'agit d'un cadre à positionnement fixe dont la dimension, perpendiculaire au sens de formatage du cadre supérieur, est fixe et dont l'autre dimension (parallèle à ce sens), est fixe ou prend implicitement la taille maximale autorisée.

Le contenu d'un cadre de ce type est extrait d'une contrainte de constitution logique du type *CommonContent* à l'aide du mécanisme de source logique.

6.3.5.23.8 RowArea (Zone ligne)

RowArea est une contrainte de constitution, qui définit un cadre composite spécifiant une zone à l'intérieur d'un cadre *TableArea*. Cette contrainte est utilisée pour présenter une ligne d'entrées dans un tableau.

Il s'agit d'un cadre à positionnement variable dont la dimension perpendiculaire au sens de formatage du cadre supérieur, est fixe et dont l'autre (parallèle à ce sens), est fixe ou spécifiée par le sous-paramètre «règle B».

Les subordonnés immédiats de cette contrainte de constitution sont composés:

- a) d'une séquence d'une ou plusieurs contraintes de constitution du type *Cell*; ou
- b) d'une séquence d'une seule contrainte de constitution du type *Cell* et d'une contrainte de constitution du type *SubRowGroup*.

6.3.5.23.9 SubRowGroup (Groupe sous-ligne)

SubRowGroup est une contrainte de constitution, qui définit un cadre composite spécifiant une zone à l'intérieur d'un cadre *RowArea*. Cette contrainte est utilisée pour présenter une ligne composite d'entrées dans un tableau.

Il s'agit d'un cadre à positionnement fixe dont la dimension perpendiculaire au sens de formatage du cadre supérieur, est fixe et dont l'autre dimension (parallèle à ce sens), peut être fixe, spécifiée par la «règle B» ou implicitement égale à la taille maximale autorisée.

Les subordonnés immédiats de cette contrainte de constitution se composent d'une séquence d'une ou de plusieurs contraintes de constitution du type *SubRow*.

6.3.5.23.10 SubRow (Sous-ligne)

SubRow est une contrainte de constitution, qui définit un cadre composite spécifiant une zone à l'intérieur d'un cadre *SubRowGroup*. Cette contrainte sert à présenter une sous-ligne d'entrées à l'intérieur d'une ligne composite dans un tableau.

Il s'agit d'un cadre à positionnement variable dont la dimension perpendiculaire au sens de formatage du cadre supérieur, est fixe et dont l'autre (parallèle à ce sens), peut être fixe, spécifiée par le sous-paramètre «règle B» ou implicitement égale à la taille maximale autorisée.

Les subordonnés immédiats de cette contrainte de constitution se composent d'une séquence de contraintes de constitution du type Cell.

6.3.5.23.11 Cell (Cellule)

Cell est une contrainte, qui définit un cadre du niveau le plus bas spécifiant une zone à l'intérieur d'un cadre RowArea ou SubRow. Cette contrainte sert à présenter une entrée dans un tableau.

Il s'agit d'un cadre à positionnement fixe dont la dimension, perpendiculaire au sens de formatage du cadre supérieur, est fixe et dont l'autre dimension (parallèle à ce sens), peut être fixe, spécifiée par le sous-paramètre «règle B» ou implicitement égale à la taille maximale autorisée.

Le contenu des cadres de ce type est extrait des contraintes de constitution logiques du type EntryElement.

6.3.6 Caractéristiques des zones d'en-tête et de bas de page

6.3.6.1 Caractéristiques générales

Les zones d'en-tête et de bas de page peuvent consister en zones de base ou en zones composites.

Une zone d'en-tête ou de bas de page de base est une zone dans laquelle le contenu est directement disposé. Ce type de zone est représenté par une contrainte de constitution applicable au constituant du type BasicHeader ou BasicFooter, respectivement.

Une zone d'en-tête ou de bas de page composite est une zone subdivisée en zones distinctes de contenu provenant d'une source et de contenu arrangé pour assurer une plus grande polyvalence en ce qui concerne la mise en page du contenu. Ce type de zone est représenté par une contrainte de constitution du type CompositeHeader ou CompositeFooter, respectivement.

Dans le cas de zones d'en-tête ou de bas de page de base, le contenu attribué à ces zones est extrait de la partie commune de la structure logique d'un document. Dans le cas de zones d'en-tête ou de bas de page composites, le contenu peut de même être extrait de la partie commune de la structure logique d'un document, mais il peut aussi être extrait du contenu commun spécifié dans la structure physique générique.

6.3.6.2 BasicHeader et BasicFooter (En-tête de base et bas de page de base)

BasicHeader et *BasicFooter* sont des contraintes de constitution, qui définissent des cadres du niveau le plus bas représentant, dans une page, les zones qui sont réservées au contenu commun.

Ces types de cadre ont des positions et des dimensions fixes. Le positionnement de ces cadres dans une page et les orientations de mise en page qui peuvent être spécifiées pour eux dépendent du type de mise en page utilisé (voir 6.3.4.5).

Le contenu qui est désigné dans ces cadres est extrait, à l'aide du mécanisme de source logique, du contenu associé aux classes d'objets logiques composites du type CommonContent.

6.3.6.3 CompositeHeader et CompositeFooter (En-tête composite et bas de page composite)

CompositeHeader et *CompositeFooter* sont des contraintes de constitution, définissant les cadres composites qui représentent, dans une page, les zones réservées au contenu commun.

Ces types de cadre ont des positions et des dimensions fixes. Le positionnement de ces cadres dans une page et les orientations de mise en page qui peuvent être spécifiées pour eux dépendent du type de mise en page utilisé (voir 6.3.4.5).

Les éléments subordonnés de ces cadres peuvent consister en l'une ou l'autre série d'éléments suivants:

- a) nombre et combinaison quelconques de cadres à positionnement variable des types SourcedContentVariable et ArrangedContentVariable; ou
- b) nombre et combinaison quelconques de cadres à positionnement fixe des types SourcedContentFixed et ArrangedContentFixed.

Dans le cas b) ci-dessus, les cadres subordonnés peuvent se chevaucher sans restriction.

6.3.6.4 SourcedContentVariable (Contenu variable provenant d'une source)

Un cadre *SourcedContentVariable* est une contrainte de constitution, qui définit un cadre du niveau le plus bas qui contient un contenu commun extrait de la structure logique générique. Ce cadre a une position variable et il est utilisé pour le contenu de positionnement qui est généré pendant le processus de mise en page, par exemple une séquence de caractères contenant un numéro de page, un titre de chapitre, etc.

Ce cadre peut être placé dans la zone de corps ainsi que dans la zone d'en-tête ou de bas de page:

- lorsque ce cadre est dans la zone d'en-tête ou de bas de page, il est le subordonné immédiat du cadre de la contrainte de constitution *CompositeHeader* ou *CompositeFooter*;
- lorsque ce cadre est dans la zone de corps, il est le subordonné immédiat du cadre de la contrainte de constitution *VariableCompositeBody*, *CompositeColumnVariable*, *CompositeColumnFixed*, *CompositeCommon*, *SnakingColumns* ou *CompositeFloat*.

SnakingColumns sert à placer common contents dans le format à plusieurs colonnes. *CompositeFloat* sert à placer common contents le long d'une figure ou d'un formulaire.

L'attribut «source logique» doit être spécifié pour ce cadre afin d'indiquer l'instance particulière de la contrainte de constitution *CommonContent* qui contient le contenu à mettre en page.

Lorsque ce cadre est un subordonné de *CompositeHeader* ou *CompositeFooter*:

- Le sens de formatage du cadre est:
 - 270° pour les mises en page d'en-tête/de bas de page A1 et A2;
 - 180° pour les mises en page d'en-tête/de bas de page B1 et B2 (voir 6.3.4.5 et l'observation au 7.4.3.21).
- La dimension horizontale du cadre est:
 - soit fixe soit spécifiée par le sous-paramètre «taille maximale» pour les mises en page d'en-tête/de bas de page A1, A2 et B2;
 - soit fixe soit spécifiée par le sous-paramètre «règle B» pour la mise en page d'en-tête/de bas de page B1;
- La dimension verticale du cadre est:
 - soit fixe soit spécifiée par le sous-paramètre «règle B» pour les mises en page d'en-tête/de bas de page A1 et A2;
 - soit fixe soit spécifiée par le sous-paramètre «taille maximale» pour la mise en page d'en-tête/de bas de page B1;
 - fixe uniquement dans le cas d'une mise en page d'en-tête/de bas de page B2.

Lorsque ce cadre est un subordonné de *VariableCompositeBody*, *CompositeColumnVariable*, *CompositeColumnFixed* ou *CompositeCommon*:

- le sens de formatage du cadre est 270°, 0° ou 180° pour les mises en page de corps A, B ou C respectivement (voir 6.3.4.5 et le commentaire au 7.4.3.21);
- la dimension du cadre dans la direction perpendiculaire à l'orientation de mise en page du cadre supérieur est soit fixe soit spécifiée par le sous-paramètre «taille maximale»;
- la dimension du cadre dans la direction parallèle à l'orientation de mise en page du cadre supérieur est soit fixe soit spécifiée par le sous-paramètre «règle B».

Lorsque ce cadre est un subordonné de *SnakingColumns*:

- le sens de formatage est 270°, 0° ou 180° pour les mises en page de corps A, B ou C respectivement (voir 6.3.4.5 et le commentaire au 7.4.3.21);
- la dimension du cadre dans la direction perpendiculaire à l'orientation de mise en page du cadre supérieur est spécifiée par le sous-paramètre «règle B» ou le sous-paramètre «taille maximale» sauf que seul le sous-paramètre «taille maximale» peut être utilisé pour la mise en page de corps C;
- la dimension du cadre dans la direction parallèle à l'orientation de mise en page du cadre supérieur est fixe.

Lorsque ce cadre est un subordonné de *CompositeFloat*:

- l'orientation de mise en page du cadre est 270°, 0° ou 180° pour des mises en page de corps A, B ou C respectivement (voir 6.3.4.5 et le commentaire au 7.4.3.21);

- la dimension du cadre dans la direction perpendiculaire à l'orientation de mise en page du cadre supérieur est fixe ou spécifiée soit par le sous-paramètre «règle B» soit par le sous-paramètre «taille maximale»;
- la dimension du cadre dans la direction parallèle à l'orientation de mise en page du cadre supérieur est soit fixe soit spécifiée par le sous-paramètre «règle B».

6.3.6.5 ArrangedContentVariable (Contenu arrangé variable)

Un cadre *ArrangedContentVariable* est une contrainte de constitution, qui définit un cadre du niveau le plus bas qui contient un contenu commun prédéfini situé dans la structure physique générique. Ce cadre a une position variable et des dimensions fixes.

Ce cadre référence un ou plusieurs pavés de type *GenericBlock* (voir 6.3.8) qui contiennent le contenu à disposer dans ce cadre. Ainsi, ce cadre est généralement utilisé quand on doit obligatoirement disposer un contenu commun prédéterminé.

6.3.6.6 SourcedContentFixed (Contenu fixe provenant d'une source)

Un cadre *SourcedContentFixed* est une contrainte de constitution, qui définit un cadre du niveau le plus bas qui contient un contenu commun extrait de la structure logique générique. Ce cadre a une position et des dimensions fixes.

Ce cadre doit obligatoirement spécifier l'attribut «source logique» qui indique le cas particulier de la contrainte applicable au constituant *CommonContent* qui contient le contenu à disposer dans ce cadre.

Ce cadre peut être placé dans la zone de corps ainsi que dans la zone d'en-tête/de bas de page:

- lorsque ce cadre est dans la zone d'en-tête/de bas de page, il est le subordonné immédiat du cadre de la contrainte de constitution *CompositeHeader* ou *CompositeFooter*;
- lorsque ce cadre est dans la zone de corps, il est le subordonné immédiat du cadre de la contrainte de constitution *FixedCompositeBody*, *CompositeCommon* ou *SynchronizedColumns*.

Lorsque ce cadre est un subordonné de *CompositeHeader* ou *CompositeFooter*:

- le sens de formatage de ce cadre est 270° pour les mises en page d'en-tête/de bas de page A1 et A2;
- le sens de formatage de ce cadre est 180° pour les mises en page d'en-tête/de bas de page B1 et B2 (voir 6.3.4.5).

Lorsque ce cadre est un subordonné de *FixedCompositeBody*, *CompositeCommon* ou *SynchronizedColumns*:

- le sens de formatage de ce cadre est 270°, 0° ou 180° pour les mises en page de corps A, B ou C respectivement (voir 6.3.4.5).

Ainsi, comme dans le cas des cadres *SourcedContentVariable*, ce cadre sert au positionnement du contenu qui est produit pendant le processus de formatage, comme une séquence de caractères contenant un numéro de page.

6.3.6.7 ArrangedContentFixed (Contenu arrangé fixe)

Un cadre *ArrangedContentFixed* est une contrainte de constitution, qui définit un cadre du niveau le plus bas qui contient un contenu commun prédéterminé extrait de la structure physique générique. La position et les dimensions de ce cadre sont fixes.

Ce cadre référence un ou plusieurs pavés du type *GenericBlock* (voir 6.3.7) qui contiennent le contenu à disposer dans ce cadre. Ainsi, ce cadre est généralement utilisé quand il est obligatoire de disposer le contenu commun dans des positions prédéterminées dans les zones d'en-tête ou de bas de page.

6.3.7 GenericBlock et SpecificBlock (Pavé générique et pavé spécifique)

Deux types de contraintes de constitution du type pavé sont définis, à savoir *GenericBlock* et *SpecificBlock*.

Les classes d'objets du type *GenericBlock* peuvent apparaître dans la structure physique générique sous forme d'éléments subordonnés aux classes d'objets des types *ArrangedContentVariable* et *ArrangedContentFixed*. Au cours du processus de mise en page visant à produire un document sous forme retraitable formatée, des pavés équivalents peuvent apparaître dans la structure physique spécifique.

Les objets du type *SpecificBlock* ne peuvent apparaître que dans la structure physique spécifique. Créés pendant le processus de mise en page du document, ils résultent de la mise en place d'objets logiques de base dans les cadres du niveau le plus bas qui constituent les zones de corps, d'en-tête et de bas de page.

6.4 Caractéristiques physiques du document

Les mécanismes de contrôle de l'attribution des constituants logiques aux différentes zones de la structure physique sont définis au 6.4.1. Les mécanismes de contrôle de la mise en page du contenu dans les zones attribuées sont définis au 6.4.2.

Ces mécanismes s'appliquent aux documents pour lesquels une structure physique générique est spécifiée. Quand une structure physique générique n'est pas présente, ces mécanismes sont limités à ceux décrits au 6.4.3.

6.4.1 Contrôle de flux

Divers mécanismes sont prévus pour contrôler l'attribution aux ensembles de pages, aux pages et aux zones de corps des contraintes de constitution qui représentent les parties «corps» de la structure logique d'un document. Ces mécanismes sont décrits aux 6.4.1.1, 6.4.1.2 et 6.4.1.3. Les mécanismes de contrôle de la mise en page des parties «communes» d'un document sont décrits au 6.4.1.4.

6.4.1.1 Attribution du contenu aux ensembles de pages

Les deux méthodes ci-après sont prévues pour l'attribution aux ensembles de pages des contraintes de constitution associées à la partie «corps» du document:

- a) mise en page dans un ensemble de pages désigné;
- b) début d'un nouvel ensemble de pages.

La première méthode permet de spécifier qu'une partie d'un document doit être mise en page entièrement à l'intérieur d'un ensemble de pages spécifié. Cette méthode peut être utilisée pour les contraintes de constitution Passage, NumberedSegment, Paragraph, Table, Figure, NumberedList, UnNumberedList et DefinitionList, utilisant l'attribut «classe d'objets physiques» qui désigne l'identificateur de classe d'objet de l'ensemble de classe de pages requis.

La seconde méthode permet de spécifier une contrainte sur les constituants logiques et d'indiquer que toutes les parties ultérieures du document devront être mises en page dans un nouvel ensemble de pages. Cette méthode peut être spécifiée pour les contraintes de constitution logiques suivantes:

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

A cette fin, on utilise l'attribut «nouvel objet physique» qui spécifie l'identificateur de classe d'objet de la classe requise de l'ensemble de pages.

6.4.1.2 Coupures de page

La coupure de page permet de spécifier les objets logiques découlant d'une contrainte de constitution logique particulière d'un document et toutes les parties ultérieures du document doivent être mises en page dans une nouvelle page. La page spécifiée doit faire partie de l'ensemble de pages dans lequel les objets logiques de la contrainte de constitution logique immédiatement précédente sont présentés. La coupure de page ne doit pas être utilisée pour présenter une partie de document dans un nouvel ensemble de pages. Si un nouvel ensemble de pages est nécessaire, il doit être expressément spécifié comme indiqué au 6.4.1.1.

Cette fonction peut être spécifiée pour des objets logiques des contraintes de constitution logiques suivantes:

- Passage
- NumberedSegment

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

La coupure de page de type d'objet est obtenue à l'aide de l'attribut «nouvel objet physique». Cet attribut peut spécifier la valeur «page», qui indique que la contrainte de constitution logique doit être présentée au début de la prochaine page disponible, qui peut être de n'importe quelle classe. Il peut aussi spécifier un identificateur d'objet indiquant la classe de page dans laquelle le constituant doit être présenté.

6.4.1.3 Attribution d'un contenu aux zones de corps

6.4.1.3.1 Introduction

Si la page à laquelle le contenu est attribué contient une zone de corps de base, le contenu est mis en page en séquence dans cette zone de corps sous la forme d'une colonne unique.

Si la page contient une zone de corps composite, c'est-à-dire un cadre VariableCompositeBody ou FixedCompositeBody, le contenu est attribué aux zones subordonnées de cette zone de corps, comme indiqué ci-dessous.

Le mécanisme général de mise en page est décrit au 6.4.1.3.2. Cependant, des services complémentaires de mise en page sont prévus pour la présentation des constituants logiques de types Figure, qui peuvent contenir des éléments de type Artwork ou Forms (voir 6.4.1.3.7), et Table (voir 6.4.1.3.8). De même, le formatage de Footnotes est décrit au 6.4.1.3.10).

6.4.1.3.2 Mécanisme général de mise en page du contenu dans une zone de corps composite

Pour la mise en page du contenu dans une zone de corps composite ayant plus d'une classe de cadre subordonné (à l'exception des classes de cadres FootnoteArea) il est nécessaire d'indiquer, directement ou indirectement, les zones possibles à utiliser.

Des objets logiques de base autres que ceux figurant à l'intérieur d'une structure de bas de page peuvent être spécifiés comme devant être présentés dans des instances d'une ou plusieurs classes de cadres du niveau le plus bas. Pour ce faire, on attribue à chaque objet logique de base une valeur de l'attribut «catégorie de présentation» qui correspond à la valeur de l'attribut «catégories autorisées» qui s'applique au cadre de niveau le plus bas dans lequel le contenu doit être présenté.

A noter que des objets logiques de base appartenant à une structure logique spécifique à laquelle cet attribut ne s'applique pas seront présentés uniquement dans le cadre de plus bas niveau qui a la valeur implicite de l'attribut «catégories autorisées».

L'utilisation de l'attribut «catégorie de mise en page» garantit que s'il n'y a pas assez de place sur une page pour présenter l'intégralité du contenu affecté à un type de zone particulier, la mise en page du contenu se poursuivra automatiquement dans le même type de zone d'une page suivante, lorsque cela sera possible. Ainsi le contenu peut couler librement d'une page à la suivante lorsque le type de formatage utilisé à la fin d'une page est le même que celui utilisé au début de la page suivante. Lorsqu'il n'est pas possible de poursuivre avec le même type de zone sur la page suivante car il y a un conflit avec d'autres directives de présentation ou car l'attribut «générateur de subordonnés» de la classe de page n'autorise pas ce choix, il peut y avoir retour arrière ou un autre type de zone peut être choisi.

Il faut garantir que le mécanisme est correctement utilisé pour la mise en page de flux de mise en page indépendants. En l'absence de directives de mise en page supplémentaires, le contenu peut être placé dans l'espace disponible d'un cadre précédent de la valeur spécifiée de l'attribut «catégories autorisées». Si telle n'est pas l'intention, on peut utiliser l'attribut «nouvel objet physique» (ou l'attribut «classe d'objet physique» dans certains cas).

L'attribut «nouvel objet physique» peut être appliqué à des composants logiques chaque fois qu'un changement de mise en page est nécessaire. Cet attribut peut spécifier l'identificateur ou le nom de catégorie correspondant à la classe de cadre requise.

Lorsque la mise en page s'effectue dans une zone de colonnes serpentes, les coupures de colonnes peuvent être indiquées en utilisant l'attribut «nouvel objet physique». Cet attribut peut spécifier l'identificateur ou le nom de catégorie du cadre correspondant à la colonne dans laquelle la mise en page se poursuit. Toutefois, seule l'utilisation du nom de catégorie garantira que l'on obtient toujours une seule coupure de colonne indépendamment de la classe de cadre utilisée.

Lorsque la mise en page s'effectue dans une zone de colonnes synchronisées, on peut utiliser les noms de catégorie pour contrôler les colonnes particulières qui seront utilisées pour mettre en page les entités logiques. Chaque colonne d'une zone de colonnes synchronisées aura une valeur différente pour l'attribut «catégories autorisées» et chaque objet logique de base à mettre en page dans cette zone aura un nom de catégorie correspondant à un nom attribué à l'une des colonnes. Les entités logiques attribuées à différentes colonnes peuvent être alignées en utilisant l'attribut «synchronisation».

Les alinéas qui suivent décrivent le mécanisme de mise en page applicable aux zones subordonnées des types de cadre énumérés ci-dessus.

6.4.1.3.3 Mise en page dans les cadres BasicFloat et BasicFixture

Dans ces cadres de niveau inférieur, la mise en page se poursuit directement jusqu'à ce que l'intervention de l'attribut «nouvel objet physique» provoque le passage à une autre classe de cadre.

6.4.1.3.4 Mise en page dans les cadres SnakingColumns (Colonnes serpentes)

Un cadre SnakingColumns est un cadre composite contenant des colonnes représentées par des cadres du niveau le plus bas ou des cadres composites.

Dans le cas de cadres inférieurs, tous les cadres peuvent avoir le même nom de catégorie de façon que le contenu puisse couler d'un cadre dans le suivant. Une coupure de colonne intervient donc naturellement lorsqu'une colonne atteint la limite imposée par le cadre supérieur; la mise en page continue alors automatiquement dans la colonne suivante.

Dans le cas de colonnes composites, les zones subordonnées sont représentées par des cadres subordonnés de types BasicFloat, CompositeFloat, CompositeFixtureVariable, TableArea et FootnoteArea. On sélectionne le type de cadre dans lequel les constituants doivent être mis en page à l'aide de l'attribut «nouvel objet physique» qui indique l'identificateur ou le nom de catégorie du cadre subordonné requis. Les constituants logiques continuent à être disposés dans le type de cadre sélectionné jusqu'à ce qu'un autre type de cadre soit choisi. De même, la mise en page se poursuit automatiquement d'une colonne dans la suivante; il est également possible d'imposer une coupure de colonne, comme indiqué au 6.4.1.3.2.

6.4.1.3.5 Mise en page dans les cadres SynchronizedColumns (Colonnes synchronisées)

Un cadre SynchronizedColumns est un cadre composite contenant des colonnes représentées par des cadres subordonnés composites ou du niveau le plus bas.

Dans le cas de cadres inférieurs, tous les cadres doivent appartenir à des catégories différentes. La présentation d'objets logiques de constituants dans les diverses colonnes est ainsi régie par les noms de catégorie spécifiés pour chacun d'entre eux. On peut également utiliser l'attribut «nouvel objet physique» à cet effet, mais cela n'est pas nécessaire.

Dans le cas de colonnes composites, les zones subordonnées sont représentées par des cadres subordonnés de types BasicFloat, CompositeFloat, CompositeFixtureVariable, TableArea et FootnoteArea.

La sélection d'une colonne composite donnée ou d'une sous-zone donnée à l'intérieur d'une colonne composite s'effectue à l'aide de l'attribut «nouvel objet physique» qui spécifie l'identificateur ou le nom de catégorie de la classe de cadre particulière requise.

6.4.1.3.6 Mise en page dans les cadres CompositeFloat

Il s'agit d'un type de cadre composite qui contient deux ou plus de deux cadres subordonnés disposés «côte à côte». Le cadre subordonné approprié est choisi selon les noms de catégorie ou à l'aide de l'attribut «nouvel objet physique» qui précise l'identificateur ou le nom de catégorie approprié du type de cadre subordonné nécessaire.

6.4.1.3.7 Mise en page des Figures

Des cadres des types CompositeFixtureVariable et CompositeFixtureFixed sont prévus spécifiquement pour la présentation de constituants logiques du type Figure. De même, des cadres des types CompositeArtwork et FormArea sont prévus pour le formatage de contraintes de constitution logiques de types Artwork et Form respectivement.

Le schéma du Tableau 2 montre l'affectation de la contrainte de constitution logique Figure et de ses subordonnés, au cadre CompositeFixtureVariable ou Composite Fixture Fixed et à ses subordonnés.

Le Tableau 2 donne une mise en correspondance entre les contraintes de constitution logiques et leurs cadres et leurs subordonnés respectifs. Il indique aussi que cette mise en correspondance est hiérarchique.

TABLEAU 2/T.506

Formatage de Figure

Contrainte de constitution logique	Classe de cadre
Figure.....>	CompositeFixtureVariable ou CompositeFixtureFixed
Artwork>	CompositeArtwork
Phrase>	BasicFixture
BodyRaster.....>	BasicFixture
BodyGeometric>	BasicFixture
Form>	FormArea
EntryGroup>	EntryGroupArea
EntryElement>	FormEntryArea
Number.....>	BasicFloat
Caption>	BasicFloat
Description>	BasicFloat
Footnote>	FootnoteArea

Par exemple, le constituant Figure doit être disposé dans un seul exemplaire du cadre CompositeFixtureVariable ou CompositeFixtureFixed. Une contrainte de constitution subordonnée du type Artwork doit être disposée dans un seul exemplaire du cadre du type CompositeArtwork à l'intérieur du cadre CompositeFixture spécifié.

De même, chaque exemplaire d'un subordonné Phrase, BodyRaster ou BodyGeometric doit être disposé dans un seul exemplaire d'un cadre subordonné du type BasicFixture. Le chevauchement de cadres BasicFixture permet de former une image composite.

De façon similaire, un constituant Form doit être disposé dans un seul exemplaire d'un cadre FormArea. Les cadres subordonnés à ce cadre FormArea, c'est-à-dire les cadres EntryGroupArea et BasicFixture recevront chacun des exemplaires uniques de constituants logiques de types EntryGroup et EntryElement respectivement.

Ce mécanisme de mise en page s'obtient en spécifiant pour le constituant Figure, l'attribut «nouvel objet physique» assorti d'une valeur indiquant l'identificateur de classes de cadre appropriées dans lesquelles ce constituant doit être disposé.

Les contraintes de constitution Number, Caption et Description (et leurs subordonnés dans le cas de Caption et Number) sont disposées dans des cadres du type BasicFloat. Pour ce faire, on utilise l'attribut «nouvel objet physique» qui indique l'identificateur ou le nom de catégorie de la classe de cadre requise. Il est possible de disposer plusieurs exemplaires des contraintes de constitution Caption, Description et Number dans tel ou tel cadre BasicFloat.

Il est possible de créer des cadres du type FootnoteArea à l'intérieur des constituants CompositeFixtureVariable ou CompositeFixtureFixed pour accepter des exemplaires de contrainte de constitution logique Footnote subordonnés aux constituants Phrase, Caption ou Description.

6.4.1.3.8 Mise en page des Tableaux

Des cadres du type TableArea sont prévus spécifiquement pour la mise en page des contraintes de constitution logiques du type Table.

Le schéma du Tableau 3 montre les relations entre la contrainte de constitution logique Table et ses subordonnés, et les cadres utilisés pour leur mise en page.

TABLEAU 3/T.506
Mise en page de tableau

Logical constituant contraint	Frame class
Table>	TableArea
Row>	RowArea
EntryElement>	Cell
TableComponent>	SubRowGroup
RowComponent>	SubRow
EntryElement>	Cell
CommonContent <>	TableHeader
CommonContent <>	TableLabel

Le Tableau 3 montre qu'il existe une correspondance hiérarchique une à une entre les contraintes de constitution logiques et les cadres correspondants. Par exemple, chaque Row est formatée dans un cadre distinct de type RowArea. Chaque TableComponent subordonné à cette Row doit être formaté dans un SubRowGroup, qui est subordonné au cadre RowArea indiqué.

Le mécanisme de mise en page est réalisé en spécifiant pour les contraintes de constitution logiques Table, Row, RowComponent et EntryElement, l'attribut «nouvel objet physique», avec une valeur indiquant l'identificateur du cadre requis de types Table, RowArea, SubRowGroup, SubRow et Cell.

Pour la contrainte de constitution logique EntryElement, le mécanisme de mise en page est réalisé de l'une des deux façons suivantes:

- si l'attribut «générateur de subordonnés» du cadre supérieur (RowArea ou SubRow) du constituant EntryElement affecté est construit à l'aide de SEquence, l'attribut «nouvel objet physique» est utilisé pour spécifier une cellule dans laquelle le contenu est mis en page. La valeur de l'attribut spécifie l'identificateur de la classe de cadre requise d'EntryElement;
- si l'attribut «générateur de subordonnés» du cadre supérieur (RowArea ou SubRow) du constituant EntryElement affecté est construit à l'aide de REPetition, la valeur de l'attribut «nouvel objet physique» indique une catégorie de nom à utiliser pour le constituant EntryElement. Dans ce cas, le nom de catégorie doit être spécifié dans l'attribut «catégorie de mise en page» par les constituants EntryText, EntryRaster ou EntryGeometric et dans l'attribut «catégories autorisées» par la cellule dans laquelle le contenu est mis en page.

Dans le cas de TableComponent, l'attribut «classe d'objets physiques» est utilisé pour spécifier que cette contrainte de constitution logique doit être formatée dans un cadre SubRowGroup.

Ce mécanisme permet de formater un tableau de sorte qu'il puisse être représenté sur deux cadres ou pages successifs, ou plus. Toutefois, un tel découpage ne peut pas se produire dans un cadre SubRowGroup.

Quand un tel découpage a lieu, les cadres TableHeader et TableLabel sont répétés en haut de chaque cadre ou page dans lequel le tableau est contenu.

Le contenu attribué aux cadres TableHeader et TableLabel est dérivé des constituants logiques de type CommonContent dans la structure logique générique utilisant le mécanisme «source logique».

6.4.1.3.9 Mise en page des formulaires

Les cadres de type FormArea sont fournis spécifiquement pour le formatage des contraintes de constitution logiques de type Form.

Le schéma du Tableau 4 montre les relations entre la contrainte de constitution logique Form et ses subordonnés, et les cadres utilisés pour le formatage de ces contraintes de constitution.

TABLEAU 4/T.506

Formatage de formulaire

Contrainte de constitution logique	Classe de cadre
Form.....>	FormArea
EntryGroup>	EntryGroupArea
EntryElement>	FormEntryArea

Le Tableau 4 montre une mise en correspondance hiérarchique, entre les contraintes de constitution logiques et les cadres associés, et leurs subordonnés respectifs.

Le mécanisme de mise en page est réalisé en spécifiant pour chaque contrainte de constitution logique l'attribut «classe d'objet physique» qui indique l'identificateur de classe d'objet d'un cadre approprié, conformément au schéma ci-dessus. Ce mécanisme ne permet pas à des cadres de type FormArea de se répartir sur deux cadres supérieurs, ou plus.

Le contenu associé aux contraintes de constitution logiques EntryElement est spécifié par un des constituants EntryText, EntryRaster ou EntryGeometric. Le formatage de ce contenu est contrôlé par les directives de formatage «décalage» et «alignement des blocs».

6.4.1.3.10 Mise en page des notes de bas de page

Les objets logiques découlant des contraintes de constitution logiques de base qui représentent le contenu d'une note de bas de page (c'est-à-dire les constituants FootnoteNumber et FootnoteText) doivent obligatoirement être associés à une zone de note qui est représentée par un cadre FootnoteArea (voir 6.3.5.9).

Cette contrainte est spécifiée à l'aide de noms de catégorie. En d'autres termes, les contraintes de constitution logiques des types FootnoteNumber et FootnoteText ainsi que les constituants physiques du type FootnoteArea doivent tous avoir pour nom de catégorie «Footnote» ou «Footnote <n>».

Plusieurs notes de bas de page peuvent être placées dans une zone de note à l'intérieur d'une zone de note de bas de page donnée. Dans ce cas, les contenus de ces notes sont disposés séquentiellement dans la zone de note de bas de page dans l'ordre dans lequel ils se lisent.

Si le contenu d'une note ne peut pas être intégralement logé dans la zone de note de bas de page, il peut couler librement dans la zone de note de bas de page suivante. Une autre solution peut consister à spécifier qu'une note doit être formatée intégralement dans une zone de note de bas de page particulière. On utilise pour ce faire l'attribut «indivisibilité».

6.4.1.4 Attribution d'un contenu aux zones d'en-tête ou de bas de page

On distingue deux sortes de zones d'en-tête ou de bas de page: les zones de base et les zones composites (voir 6.3.6.1). Dans le cas d'une zone de base, le cadre qui représente cette zone spécifie l'attribut «source logique» qui spécifie l'instance particulière de la contrainte de constitution logique du type CommonContent qui doit être formatée dans cette zone. Les contraintes de constitution logiques de base subordonnées à CommonContent sont alors mises en page conformément à leur ordre séquentiel.

Dans le cas d'une zone d'en-tête ou de bas de page composite (voir 6.3.6.3), la zone est divisée en une ou plusieurs zones distinctes représentées chacune par un cadre de niveau le plus bas. Le contenu attribué à ces zones distinctes peut être extrait d'une ou de deux sources. C'est-à-dire que le contenu peut être prédéfini et représenté par un ou plusieurs pavés directement associés au cadre de niveau le plus bas. Le cas échéant, le cadre de niveau le plus bas peut spécifier l'attribut «source logique» qui, comme précédemment, indique l'entité logique particulière du type CommonContent qui doit être formatée dans ce cadre.

6.4.2 Mise en page du contenu du document

Diverses contraintes peuvent être spécifiées pour contrôler la mise en page du contenu dans les zones de corps, d'en-tête et de bas de page. Ces contraintes sont décrites ci-dessous.

6.4.2.1 Marges

Les marges sont les distances minimales, ou décalages, entre une partie du contenu du document et le bord de la zone dans laquelle le contenu est formaté. Les marges délimitent la zone maximale de la surface disponible dans laquelle le contenu peut être situé.

Des marges peuvent être spécifiées pour toutes les entités logiques de base; des valeurs de marge différentes peuvent être spécifiées pour des entités logiques de base différentes sans restriction.

Quatre marges distinctes peuvent être spécifiées pour chaque entité logique, à savoir:

- la marge de bord de fuite;
- la marge de bord d'attaque;
- la marge droite;
- la marge gauche.

Ces marges sont définies par rapport à l'orientation de mise en page spécifiée pour le cadre dans lequel le contenu doit être présenté (voir la Figure 18).

Toute combinaison des marges ci-dessus peut être spécifiée pour une entité logique particulière. Ces marges sont spécifiées par l'attribut «décalage». Toute valeur peut être spécifiée en unités de mesure de base (BMU). Si une marge particulière n'est pas spécifiée, on admet que sa valeur est de 0 BMU.

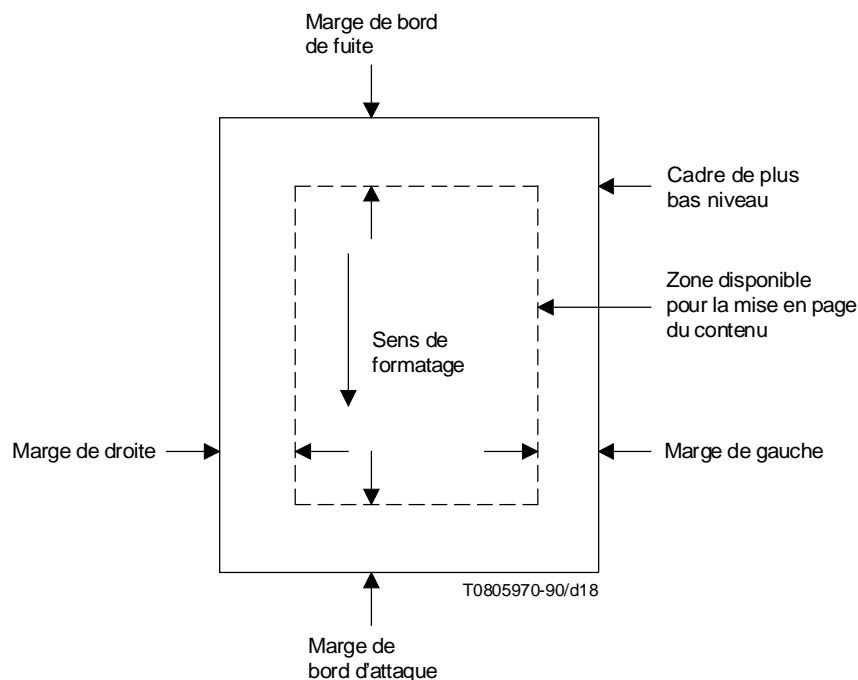


FIGURE 18/T.506
Spécification des marges

6.4.2.2 Séparation

La séparation «avant» est la distance minimale entre un objet logique et le suivant lorsqu'ils sont formatés. La séparation «arrière» est la distance minimale entre un objet logique de base et le précédent, s'il existe, lorsqu'ils sont formatés. Elles peuvent être spécifiées pour toute contrainte de constitution logique de base. Ces distances sont spécifiées en BMU par l'attribut «séparation». Si aucune valeur n'est spécifiée, on admet que la valeur de la distance minimale est de 0 BMU.

6.4.2.3 Indivisibilité

L'indivisibilité permet de spécifier si un objet logique découlant d'une contrainte de constitution logique de base ou composite peut ou non être formaté sur plusieurs pages ou sur plusieurs zones d'une même page. L'indivisibilité peut être spécifiée pour les contraintes de constitution des types Passage, NumberedSegment, Number, Title, Paragraph, Figure, Table, UnNumberedList, DefinitionList, FootnoteReference et BodyText. L'attribut «indivisibilité» est utilisé pour spécifier cette caractéristique.

6.4.2.4 Même objet physique

L'attribut «même objet physique» permet de spécifier que le début du contenu associé à un objet logique et la fin du contenu associé à l'objet logique précédent seront disposés dans un seul objet physique. Cet attribut peut être spécifié pour les objets logiques des types NumberedSegment, Paragraph, Number, Title, Figure Table, UnNumberedList, NumberedList, DefinitionList, Footnote, FootnoteReference, BodyText, BodyRaster et BodyGeometric. L'attribut «même objet physique» est utilisé pour spécifier cette caractéristique.

6.4.2.5 Concaténation

La concaténation permet de spécifier que le contenu associé à un objet logique découlant d'une contrainte de constitution logique de base et le contenu associé à l'objet logique découlant de la contrainte de constitution logique de base précédente sont à considérer comme un flot ininterrompu de contenu. La concaténation peut être spécifiée pour les contraintes de constitution des types BodyText, Number, ReferencedContent, FootnoteReference, FootnoteNumber, FootnoteText, CommonText et PageNumber. L'attribut «concaténation» est utilisé pour spécifier cette caractéristique.

6.4.2.6 Alignement de pavés

L'alignement de pavés permet de spécifier le contenu associé à une entité logique de base comme étant 'aligné à gauche', 'aligné à droite' ou 'centré' dans la zone dans laquelle ce contenu est mis en page. Aligné à gauche signifie que le contenu est mis en page en position adjacente à la marge gauche. Aligné à droite signifie que le contenu est mis en page en position adjacente à la marge droite et centré signifie que le contenu est disposé à mi-chemin entre les marges gauche et droite.

Cette caractéristique ne peut être spécifiée qu'à l'aide de l'attribut «alignement de pavés» pour les contraintes de constitution des types BodyText, EntryText, Number, CommonNumber, PageNumber, TableNumber, FootnoteNumber, FootnoteText, FootnoteReference, CommonReference, ReferenceContent, CurrentInstance et CommonText, lorsque celles-ci renferment un contenu de caractères formaté, BodyRaster et BodyGeometric, EntryRaster, EntryGeometric, CommonRaster et CommonGeometric.

6.4.3 Commandes de mises en page applicables en l'absence de structure physique générique

Dans les documents de forme retraits, la structure physique générique est facultative. Si la structure physique générique est omise, il appartient alors au destinataire de définir une structure physique appropriée. Il n'est pas imposé de limitations en ce qui concerne la structure physique qui est utilisée.

Lorsqu'une structure physique générique n'est pas spécifiée dans un document de forme retraits, des restrictions sont alors imposées quant aux fonctions de commande de mise en page décrites aux 6.4.1 et 6.4.2 qui peuvent être spécifiées dans le document. Ces restrictions sont indiquées ci-après.

- a) Il n'est pas possible de spécifier que certaines parties logiques d'un document doivent être attribuées à un ensemble de pages donné ou qu'une partie d'un document doit être formatée au début d'un nouvel ensemble de pages, comme indiqué au 6.4.1.1.

- b) Il est possible de spécifier des coupures de page telles que celles qui sont définies au 6.4.1.2 mais il est seulement possible d'indiquer que la mise en page doit commencer sur une nouvelle page. Il n'est pas possible de spécifier une classe de page particulière.
- c) Les parties logiques du document qui sont destinées à être mises en page dans la partie corps de texte et dans les zones en-tête/bas de page de chaque page peuvent être différenciées à l'aide des commentaires d'application (voir 6.6.4). Une exception est toutefois à signaler: il n'est pas possible de déterminer si le contenu commun doit être placé dans une zone en-tête ou dans une partie bas de page (ou divisé entre les deux).
- d) Il n'est pas possible d'indiquer le type de zone de formatage à utiliser pour formater chaque constituant logique de la partie corps de texte d'un document. En d'autres termes, il n'est pas possible d'indiquer s'il faut utiliser des zones à une colonne ou à plusieurs colonnes (voir 6.4.1.3.1). C'est au destinataire qu'il incombe d'en décider.
- e) Les notes de bas de page incorporées dans la partie corps de texte d'un document peuvent être différenciées au moyen de l'attribut «commentaires d'application». Les notes de bas de page sont destinées à être lues et formatées séparément des autres constituants logiques de la partie de corps (voir 6.4.1.3.2). Toutefois, il incombe au destinataire de décider de la manière dont les notes de bas de page doivent être mises en page.
- f) Les attributs marges, séparation, indivisibilité, même objet physique, concaténation et alignement de pavés, tels qu'ils sont définis au 6.4.2, peuvent tous être spécifiés à une seule restriction près. L'indivisibilité (voir 6.4.2.3) peut être utilisée pour spécifier qu'une contrainte applicable à un constituant logique ne doit pas être répartie sur plusieurs pages; toutefois, l'indivisibilité ne peut pas être spécifiée pour d'autres types de zones de mise en page tels que les zones à une seule colonne ou à plusieurs colonnes.

6.5 Mise en page du contenu et caractéristiques de restriction

Un document peut comporter un contenu caractères, un contenu graphique en points et un contenu graphique géométrique.

Les architectures de contenu qui peuvent être spécifiées à l'aide de l'attribut «classe d'architectures de contenu» peuvent comporter cinq types de contenus: les contenus caractères de forme formatée, les contenus caractères de forme retraitable, les contenus caractères de forme retraitable formatée, les contenus graphiques en points de forme retraitable formatée et les contenus graphiques géométriques de forme retraitable formatée. L'une quelconque de ces architectures peut être spécifiée comme la valeur par défaut dans le profil d'un document.

6.5.1 Contenu de caractères

6.5.1.1 Introduction

Le présent paragraphe définit les caractéristiques qui sont applicables au contenu de caractères d'un document ainsi que les attributs de présentation et les fonctions de commande qui peuvent être utilisés pour spécifier ces caractéristiques. Ces caractéristiques peuvent s'appliquer aux composants logiques de base et aux composants physiques, sauf indication contraire.

Peuvent être spécifiées dans le profil d'un document les valeurs par défaut des attributs de présentation suivants:

- jeux de caractères graphiques;
- sous-répertoire de caractères graphiques;
- annonceurs d'extension de code;
- espacement entre lignes;
- espacement des caractères;
- trajet des caractères;
- progression des lignes;
- orientation des caractères;
- rendu (reproduction) graphique, comportant les valeurs de paramètre suivantes: rendu par défaut, trait gras, italique, souligné, biffé, police primaire, 1^{re} police de remplacement, 2^e police de remplacement, 3^e police de remplacement, 4^e police de remplacement, 5^e police de remplacement, 6^e police de remplacement, 7^e police de remplacement, 8^e police de remplacement, 9^e police de remplacement, soulignement double, intensité normale, non souligné, non biffé;
- tableau de présentation des lignes;

- tabulation;
- indentation;
- alignement;
- décalage de la première ligne;
- itémisation;
- taille-veuve;
- taille-orpheline;
- polices de caractères;
- décalage de crénage;
- espacement ligne proportionnel;
- décalage initial.

La spécification dans un document d'une caractéristique non essentielle par un attribut de présentation ou une fonction de commande doit être indiquée dans le profil du document.

6.5.1.2 Classes d'architectures de contenu de caractères

Les documents de formes retraits et retraits formatés peuvent comporter un contenu de caractères de formes retraits, formatés ou retraits formatés. Les documents de forme formatés peuvent comporter un contenu de caractères de formes formatés et retraits formatés.

En cas d'utilisation d'un contenu de caractères, un nombre quelconque de portions de contenu peut être associé à un composant de base.

L'information de contenu d'une portion de contenu peut être absente, cela afin de permettre la représentation et l'échange de documents dans lesquels des parties du contenu peuvent être fournies, par exemple, pendant l'édition subséquente.

6.5.1.3 Répertoires de caractères

Le répertoire de caractères de base admis par le présent profil est composé des 94 caractères de la VIR de la Norme ISO 646 (révisée 1991), et du caractère espace.

Tout autre jeu de caractères graphiques enregistré conformément à la Norme ISO 2375 peut être désigné et appelé en tout point du document, à condition que son utilisation soit indiquée dans le profil du document comme étant une valeur non essentielle, à l'aide des caractéristiques de présentation des caractères «jeux de caractères graphiques». Aucune fonction d'inversion avec verrouillage n'est spécifiée dans ces caractéristiques de présentation.

Les techniques d'extension de code autorisées pour la désignation et l'appel de jeux de caractères des côtés gauche et droit du tableau de code à 8 bits (GL et GR respectivement) sont définies au 6.5.1.4.

En cas d'utilisation de ces techniques d'extension de code, les jeux de caractères graphiques désignés et/ou appelés au début d'une portion de contenu comportant un contenu de caractères sont spécifiés par l'attribut de présentation «jeux de caractères graphiques». En outre, les jeux de caractères peuvent être modifiés en un point quelconque d'une portion de contenu.

Les jeux de caractères graphiques qui s'appliquent implicitement aux portions de contenu d'un document peuvent être spécifiés dans le profil du document à l'aide de l'attribut de présentation «jeux de caractères graphiques».

Si le jeu de caractères défini dans la Norme ISO 6937-2 avec ou sans Addendum 1 est désigné et appelé, l'utilisation de tout sous-répertoire enregistré conformément à la Norme ISO 735-0 peut être spécifiée à l'aide de l'attribut de présentation «sous-répertoire de caractères graphiques». Tous les sous-répertoires étant non essentiels, leur utilisation doit être indiquée dans le profil du document. Le sous-répertoire ne doit pas être remplacé par un autre en aucun point de la portion de contenu.

NOTES

1 Le répertoire de caractères de base admis par le présent profil ne correspond pas à la valeur par défaut normalisée spécifiée dans la Rec. T.416 du CCITT | ISO 8613-6; par conséquent, il est peut-être nécessaire de spécifier, dans le profil d'un document donné, qu'il s'agit là de la valeur par défaut utilisée pour ce document.

2 Les Recommandations T.50 et T.51 révisées ainsi que la nouvelle Recommandation T.52 sont en cours de préparation. Les Recommandations T.50 et T.51 sont censées être entièrement compatibles avec les Normes ISO 646 (révisées 1991) (ISO-IR 6) et ISO 6937 (en cours de révision) respectivement.

6.5.1.4 Techniques d'extension de code

Les techniques d'extension de code spécifiées dans la Norme ISO 2022 peuvent être utilisées sous réserve des restrictions suivantes:

- a) *Jeu G0* – seules ISO-IR6 (VIR de la Norme ISO 646 révisée 1991), ISO-IR2 (jeu primaire de la Norme ISO 6937-2) et toute autre version de ISO 646 (révisée 1991) peuvent être désignées pour cet ensemble; ces jeux de caractères graphiques ne peuvent être appelés qu'en GL.
- b) *Jeux G1, G2 et G3* – il n'est pas imposé de restrictions en ce qui concerne les jeux de caractères qui peuvent être désignés pour ces jeux; ces jeux de caractères graphiques ne peuvent être appelés qu'en GR.
- c) Les fonctions d'inversion avec verrouillage et de positionnement unique de clavier autorisées sont les suivantes:
 - LS0 pour appeler le jeu G0 en GL;
 - LS1R pour appeler le jeu G1 en GR;
 - LS2R pour appeler le jeu G2 en GR;
 - LS3R pour appeler le jeu G3 en GR;
 - SS2 pour appeler un caractère du jeu G2 en GL;
 - SS3 pour appeler un caractère du jeu G3 en GL.

GL et GR désignent ici respectivement les parties gauche et droite du tableau de code à 8 bits.

- d) Lors de la spécification de l'attribut de présentation «jeux de caractères graphiques», il est nécessaire d'appeler des jeux de caractères tant pour GL que pour GR. Ainsi, un jeu de caractères autorisé doit être désigné en G0 [voir le point a) ci-dessus] et appelé en GR. Il est également nécessaire d'appeler en GR un jeu de caractères qui a été désigné dans un jeu G1, G2 ou G3.
- e) Le jeu vide doit être désigné en G1 et appelé en GR si aucun autre jeu de caractères spécifique n'est appelé en GR.

Les techniques d'extension de code autorisées sont illustrées aux Figures 19 et 20.

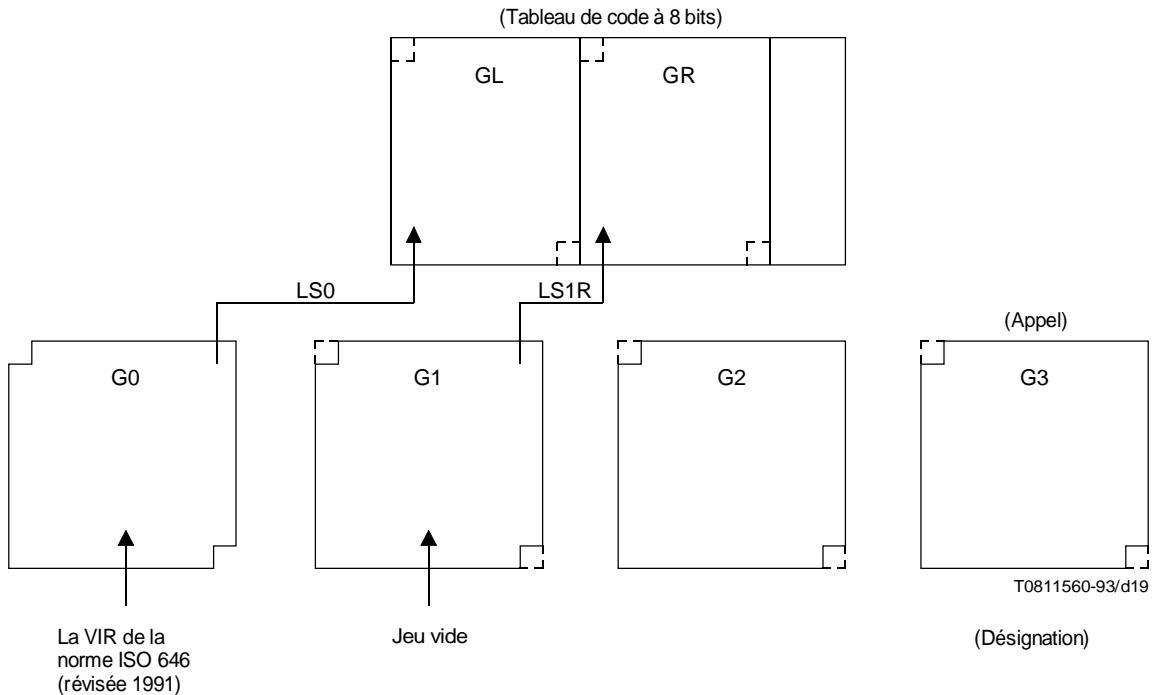


FIGURE 19/T.506

Caractéristiques d'extension de base (jeux de caractères de base)

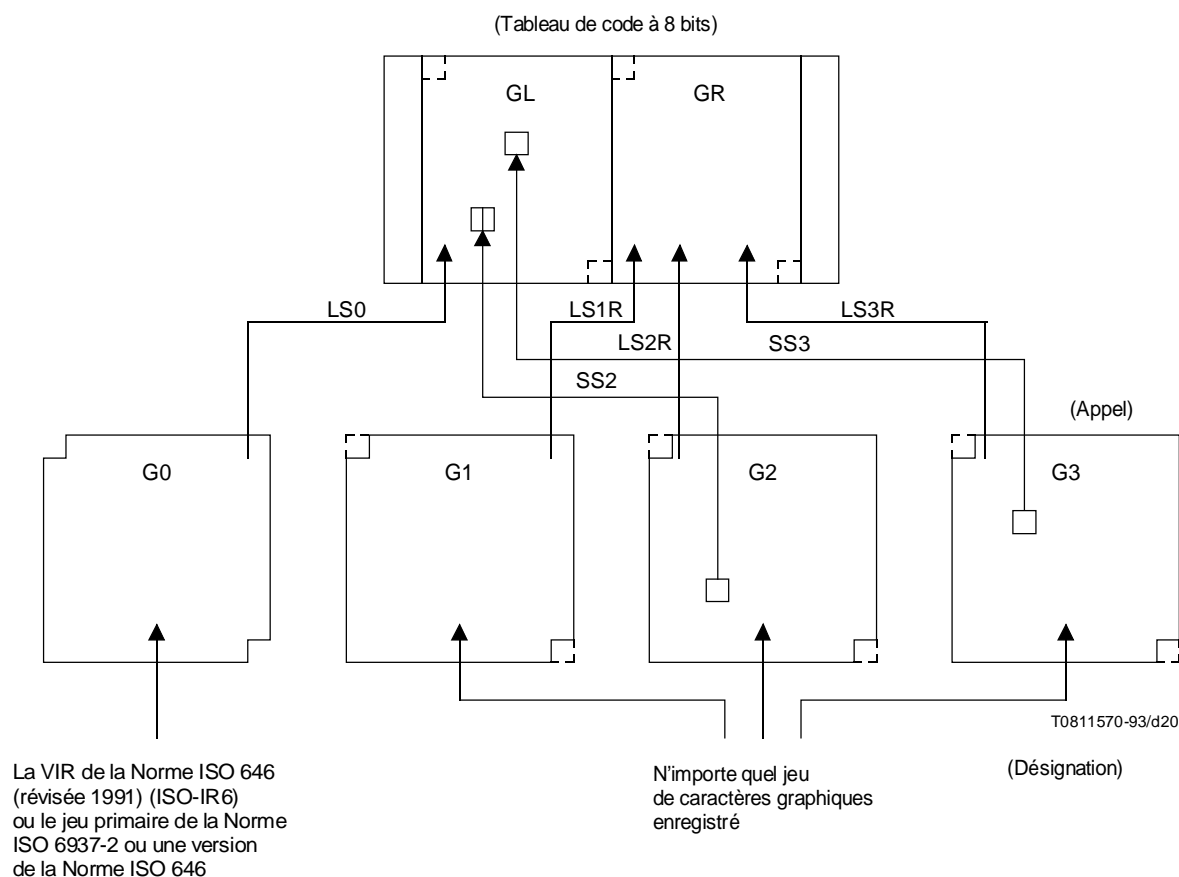


FIGURE 20/T.506

Caractéristiques d'extension de code (pour tous les jeux de caractères possibles)

L'annonce et le codage de ces fonctions doivent suivre les modalités spécifiées dans la Norme ISO 2022.

Les techniques d'extension de code qui sont utilisées ou peuvent être utilisées dans un composant de base doivent être spécifiées par l'attribut de présentation «annonceurs d'extension de code». Les annonceurs d'extension de code par défaut utilisés tout au long d'un document peuvent être spécifiés dans le profil du document, également à l'aide de l'attribut de présentation «annonceurs d'extension de code».

NOTE – Conformément à la Rec. T.416 du CCITT | ISO 8613-6, aucune restriction n'est imposée concernant le nombre de jeux de caractères graphiques qui peuvent être désignés et (ou) appelés dans l'attribut de présentation «jeux de caractères graphiques», pour autant que les restrictions définies ici soient appliquées. Par conséquent, la désignation en un jeu G donné prévaut sur une désignation précédente en ce jeu et l'appel en GL ou GR prévaut sur l'appel précédent en GL ou GR respectivement. L'ordre de succession des séquences de désignation et (ou) d'appel dans l'attribut «jeux de caractères graphiques» est donc important.

6.5.1.5 Espacement entre les lignes

N'importe quelle valeur d'espacement ligne peut être spécifiée. Les valeurs de 150, 200, 300 et 400 BMU sont les valeurs de base; l'utilisation de toute autre valeur dans un document étant non essentielle, elle doit être indiquée dans le profil du document.

L'espacement ligne peut être spécifié au début du contenu associé à un composant de base à l'aide de l'attribut de présentation «espacement ligne». Cette valeur peut être modifiée en tout point de la portion de contenu à l'aide des fonctions de commande SVS (*select line spacing*) et SLS (*set line spacing*).

6.5.1.6 Espacement des caractères

Toute valeur de l'espacement des caractères peut être spécifiée comme valeur de base.

L'espacement des caractères peut être spécifié au début du contenu associé au composant de base, à l'aide de l'attribut de présentation «espacement des caractères». Cette valeur peut être modifiée en tout point de la portion de contenu à l'aide des fonctions de commande SHS (*select character spacing*) et SCS (*set character spacing*).

NOTE – Les paramètres SHS de valeur 0, 1, 2, 3 et 4 sont actuellement fournis. L'utilisation des paramètres 5 et 6 est actuellement à l'étude en vue d'applications aux caractères chinois.

6.5.1.7 Trajet des caractères et progression des lignes

Les deux sens d'écriture horizontale et verticale peuvent être utilisés dans un document. Dans le cas de l'écriture horizontale, les caractères progressent de gauche à droite ou de droite à gauche de la page et la progression des lignes se fait de haut en bas de la page. Dans le cas de l'écriture verticale, les caractères progressent de haut en bas de la page et la progression des lignes se fait de droite à gauche. L'utilisation de ces sens d'écriture est limitée par le type de mise en page utilisé.

Pour la mise en page A, seule l'écriture horizontale peut être utilisée dans la zone. Par conséquent, dans ce cas, les valeurs spécifiées pour le trajet des caractères et la progression des lignes sont respectivement de 0 et 270 degrés ou de 180 et 90 degrés.

Pour la mise en page de corps B, là encore seule l'écriture horizontale peut être utilisée dans la zone de corps. Toutefois, dans ce cas, le contenu qui figure dans la zone de corps est présenté pour être vu lorsque la page est orientée horizontalement (paysage) et le contenu qui figure dans les zones d'en-tête et de bas de page est présenté pour être vu lorsque la page est orientée verticalement (portrait). Ainsi, pour la mise en page B, dans la zone de corps, les valeurs spécifiées pour le sens des caractères et la progression des lignes sont respectivement de 90 et 270 degrés ou de 270 et 90 degrés.

Pour la mise en page C, seule l'écriture verticale peut être utilisée dans la zone de corps. Ainsi, dans la zone de corps, les valeurs spécifiées pour le trajet des caractères et la progression des lignes sont respectivement de 270 et 270 degrés.

Quant aux zones d'en-tête et de bas de page, seule l'écriture horizontale peut être utilisée si ces zones sont situées au-dessus et en dessous de la zone de corps. Ainsi dans ce cas, les valeurs spécifiées pour le sens des caractères et la progression des lignes sont respectivement de 0 et 270 degrés ou de 180 et 90 degrés (voir la Figure 2).

Si les zones d'en-tête et de bas de page sont situées à gauche et à droite de la zone de corps, seule l'écriture horizontale peut être utilisée. Les valeurs spécifiées pour le trajet des caractères et la progression des lignes sont alors de 270 degrés dans les deux cas (voir la Figure 4).

Toutes les valeurs du trajet des caractères et de la progression des lignes sont essentielles. Les valeurs du trajet des caractères et de l'espacement ligne peuvent être spécifiées au début du contenu associé à un composant de base, en utilisant les attributs de présentation «sens (d'écriture) des caractères» et «progression des lignes». Ces valeurs ne peuvent pas être modifiées à l'intérieur d'une partie de contenu.

6.5.1.8 Commandes de positionnement des caractères

La position active d'un caractère (définie dans la Rec. T.416 du CCITT | ISO 8613-6) peut être déplacée vers l'avant ou vers l'arrière dans le sens de la progression des lignes à l'aide des fonctions de commande VPB et VPR. Ces fonctions de commande peuvent être spécifiées dans toutes les formes de contenu de caractères et il est possible de spécifier n'importe quelle valeur de paramètre.

La position active d'un caractère peut être déplacée vers l'avant ou vers l'arrière dans le sens de l'orientation des caractères à l'aide des fonctions de commande HPB et HPR.

Il est possible d'augmenter ou de réduire l'espacement entre les caractères en utilisant respectivement les fonctions de commande SACS et SRCS. Il est aussi possible de fixer la largeur du caractère ESPACE en utilisant la fonction SSW.

Les fonctions de commande HPB, HPR, SACS, SRCS et SSW ne peuvent être spécifiées que dans un contenu de caractères formaté; il est possible de spécifier n'importe quelle valeur de paramètre.

6.5.1.9 Orientation des caractères

La valeur spécifiée pour l'orientation des caractères peut être 0, 90, 180 ou 270°.

Les valeurs de 0, 90 et 180° spécifiées pour l'orientation des caractères sont généralement fonction de l'écriture. Si l'on utilise l'écriture horizontale les caractères ne peuvent être orientés qu'à 0°. Si l'on utilise l'écriture verticale les caractères peuvent être orientés à 90° ou 180°.

Toutes les valeurs de l'orientation des caractères sont de base. La valeur de l'orientation des caractères est spécifiée au début du contenu associé à un composant de base par l'attribut de présentation «orientation des caractères». Cette valeur ne peut pas être modifiée à l'intérieur du contenu.

6.5.1.10 Mise en relief

Les modes ci-après de mise en relief des caractères graphiques peuvent être distingués:

- reproduction par défaut;
- intensité normale;
- intensité accrue (gras);
- italique;
- sans italique;
- souligné;
- double soulignement;
- non souligné;
- biffé;
- non biffé;
- clignotement lent;
- clignotement rapide;
- pas de clignotement;
- image négative;
- image positive.

Tous les modes ci-dessus sont des modes de mise en relief de base. Si aucun mode par défaut n'est expressément spécifié dans le profil du document, le mode par défaut est alors la reproduction par défaut.

Le mode de mise en relief peut être spécifié au début du contenu associé à un composant de base, à l'aide de l'attribut de présentation «reproduction graphique». Ce mode peut être remplacé par un autre en tout point du contenu, à l'aide de la fonction de commande SGR.

Le mode de mise en relief demeure actif en tout point du contenu associé à un composant de base jusqu'à ce qu'il soit remplacé par un mode mutuellement exclusif ou par la spécification 'reproduction par défaut'. Les modes mutuellement exclusifs sont: intensité normale/accrue, italique/sans italique, souligné/non souligné, biffé/non biffé et image positive/négative. Un mode de chaque doublet de modes mutuellement exclusifs peut être actif en tout point du contenu du document.

La reproduction par défaut supprime les effets de toutes les méthodes de mise en relief actives et spécifie que le texte doit être reproduit sur la base des paramètres de reproduction par défaut du système de reproduction utilisé. Par exemple, s'il faut avoir la certitude que le contenu ne sera pas souligné, il est nécessaire de spécifier expressément que le mode souligné ne doit pas être utilisé.

6.5.1.11 Tabulation

Des positions d'arrêt de tabulation peuvent être spécifiées sur toute position de caractère le long du trajet des caractères. Chaque point d'arrêt est spécifié comme suit:

- a) position de tabulation par rapport à la position de marge dans la direction opposée au trajet des caractères;
- b) qualificateur facultatif d'alignement, spécifiant le type d'alignement à utiliser sur la position de tabulation désignée. Le type peut être spécifié comme l'un des suivants:
 - aligné à gauche;
 - aligné à droite;
 - centré;
 - aligné complètement.

Ces qualificateurs d'alignement sont définis dans la Rec. T.416 du CCITT | ISO 8613-6. Si le qualificateur d'alignement n'est pas expressément spécifié, on admet alors que le type d'alignement à utiliser est l'alignement à gauche.

Un seul ensemble de points d'arrêt de tabulation peut être spécifié comme étant applicable au contenu associé à un composant de base. Le nombre d'arrêts de tabulation qui peuvent être spécifiés dans un ensemble donné n'est pas limité.

L'ensemble de positions d'arrêt de tabulation associé au contenu d'un composant de base est spécifié à l'aide de l'attribut de présentation «table de formatage des lignes». Les positions d'arrêt de tabulation sont appelées en tout point du contenu à l'aide de la fonction de commande STAB.

Les numéros de référence de tabulation utilisés dans la fonction de commande STAB et l'attribut de présentation associé «tableau de formatage des lignes» sont choisis de façon que, dans tout tableau de formatage des lignes, les numéros de référence soient uniques, séquentiels dans la direction du trajet des caractères et ne contiennent pas de zéros non significatifs.

6.5.1.12 Indentation

L'indentation est la distance entre le premier caractère d'une ligne de contenu et la position de marge en sens opposé au trajet des caractères. La valeur d'indentation spécifiée détermine donc la valeur de la position de départ de ligne (définie dans la Rec. T.416 du CCITT | ISO 8613-6.

L'indentation modifie temporairement la position de décalage du texte en sens opposé au trajet des caractères. Lorsque le texte est formaté, la disposition se fait entre la position d'indentation et la position de marge dans le sens du trajet des caractères.

Toute valeur d'indentation peut être spécifiée pour les composants logiques de base, à l'aide de l'attribut de présentation «indentation». La valeur d'indentation ne peut être modifiée en aucun point d'une portion de contenu.

6.5.1.13 Alignement

Cette caractéristique concerne la manière dont les premier et dernier caractères de chaque ligne de contenu de caractères doivent être formatés au cours du processus de formatage.

Les valeurs d'alignement ci-après peuvent être spécifiées comme étant les valeurs d'alignement de base:

- aligné à gauche;
- aligné à droite;
- centré;
- justifié.

La sémantique de ces valeurs est définie dans la Rec. T.416 du CCITT | ISO 8613-6.

L'attribut de présentation «alignement» est utilisé pour spécifier l'alignement applicable au contenu associé à un composant de base. La valeur d'alignement ne peut être modifiée en aucun point d'une portion de contenu.

6.5.1.14 Format de la première ligne

Cette caractéristique spécifie la disposition de la première ligne du contenu associé à un composant de base et permet l'itémisation des paragraphes.

Elle permet de placer le premier caractère du contenu en un point du trajet des caractères, par rapport à la position d'indentation (définie au 6.5.1.12). Ce point peut être pris dans le sens du trajet des caractères ou en sens inverse, par rapport à la position d'indentation.

En outre, cette caractéristique permet la spécification d'un identificateur d'article sur la première ligne. L'identificateur d'article consiste en une chaîne de caractères qui précède les caractères restants qui forment la première ligne et qui est séparée de ces caractères. La fonction de commande CR (retour chariot) est employée comme séparateur.

Les caractéristiques prévues correspondent aux exemples 10.1 à 10.5 présentés à la Figure 10 de la Rec. T.416 du CCITT | ISO 8613-6.

Le format de la première ligne est spécifié par les attributs de présentation «décalage de la première ligne» «indentation» et «itémisation». Seules les valeurs d'attributs qui servent à former les exemples montrés à la Figure 10 de la Rec. T.416 du CCITT | ISO 8613-6 peuvent être utilisées.

6.5.1.15 Taille-veuve et taille-orpheline

La taille-veuve spécifie le nombre minimal de lignes du contenu à attribuer au cadre suivant ou à la page suivante lorsque le contenu associé à un composant logique de base est formaté de telle sorte qu'il déborde sur deux cadres ou deux pages. En pareil cas, il peut être nécessaire de transférer plusieurs lignes de contenu d'un cadre ou d'une page au cadre suivant ou à la page suivante.

La taille-orpheline spécifie le nombre minimal de lignes du contenu à placer dans le cadre ou la page en cours de formatage lorsque le contenu associé à un composant logique de base est formaté sur deux cadres ou pages. Si ce nombre minimal de lignes ne peut pas être logé sur le cadre ou la page en cours de formatage, il faut alors placer l'intégralité du contenu sur le cadre suivant ou la page suivante.

Toute valeur de taille-veuve ou de taille-orpheline peut être spécifiée à l'aide des attributs de présentation respectifs «taille-veuve» et «taille-orpheline».

La taille-veuve et la taille-orpheline ne peuvent être spécifiées que pour un contenu de caractères placé dans la zone de corps des pages.

6.5.1.16 Polices de caractères

Un nombre quelconque de polices peut être utilisé dans un document. Les polices utilisées dans un document particulier sont spécifiées dans le profil de ce document à l'aide de l'attribut «liste de polices de caractères».

De plus amples renseignements concernant la spécification des références de polices dans le profil du document sont donnés dans B.2.

Les polices qui peuvent être utilisées dans le contenu associé à chaque composant de base sont spécifiées par l'attribut de présentation «polices de caractères». Jusqu'à dix polices tirées de la liste spécifiée par l'attribut «liste de polices» peuvent être spécifiées par l'attribut «polices de caractères».

La police à utiliser au début du contenu associé à un composant de base est spécifiée à l'aide de l'attribut «reproduction graphique». Les polices utilisées dans le contenu peuvent être modifiées à l'aide de la fonction de commande SGR.

Le profil du document peut spécifier, à l'aide de l'attribut «polices de caractères», un jeu par défaut comportant au maximum dix polices applicables à l'ensemble du document.

6.5.1.17 Chaînes de caractères inverses

La présente Recommandation admet l'écriture dans les deux sens. Par conséquent, on peut spécifier qu'une chaîne de caractères d'une partie de contenu associée à un composant de base soit restituée dans le sens inverse de la chaîne de caractères immédiatement précédente. De telles chaînes peuvent être spécifiées par la fonction de commande SRS définie dans la Rec. T.416 du CCITT | ISO 8613-6.

Cette fonction de commande est prévue pour les cas où le texte relève de langages différents et où le contenu des caractères est écrit, par exemple, de gauche à droite ou de droite à gauche sur la même ligne de caractères, selon le langage et/ou le jeu de caractères utilisés.

6.5.1.18 Décalage de crénage

Une valeur de décalage de crénage pour le contenu associé à un composant de base peut être spécifiée à l'aide de l'attribut de présentation «décalage de crénage». Il est nécessaire de spécifier une telle valeur lorsque certaines polices sont appelées, de manière qu'aucune partie des images de caractères ne soit placée en dehors de la zone disponible.

6.5.1.19 Espacement ligne proportionnel

L'utilisation de l'espacement ligne proportionnel peut être appelée pour le contenu associé à un composant logique de base à l'aide de l'attribut «espacement ligne proportionnel». En cas d'appel de cet attribut, l'espacement ligne est déterminé d'après les attributs associés à la police utilisée; il peut varier d'une ligne à la suivante. Ce processus dépend de l'application.

6.5.1.20 Exposants et indices

Des exposants et des indices peuvent être spécifiés en tout point du contenu associé à un composant de base, à l'aide des fonctions de commande PLU (*partial line up*) et PLD (*partial line down*). L'utilisation de ces fonctions de commande doit être conforme à la Rec. T.416 du CCITT | ISO 8613-6.

6.5.1.21 Coupure de ligne

Les fonctions de commande BPH (*break permitted here*) et NBH (*no break here*) peuvent être insérées dans le contenu de caractères de forme retraitable pour indiquer respectivement, lors du formatage du contenu, les points où il peut et où il ne peut pas y avoir coupure de ligne.

6.5.1.22 Remplacement de caractères

La fonction de commande SUB est prévue pour représenter des caractères produits par un système local qui ne peuvent être représentés par aucun des caractères des jeux de caractères admis par la présente Recommandation.

6.5.1.23 Point initial

Le point initial, qui est applicable à des composants physiques de base, peut être spécifié par l'attribut «décalage initial». Toute valeur peut être spécifiée.

6.5.1.24 Utilisation des fonctions de commande

On trouvera ci-après la liste de toutes les fonctions de commande et valeurs de paramètre (lorsqu'il y a lieu) qui peuvent être spécifiées dans le contenu caractères:

SHS	choix de l'espacement des caractères (<i>select character spacing</i>) (valeurs de paramètre autorisées: 0, 1, 2, 3, 4)
SCS	établissement de l'espacement des caractères (<i>set character spacing</i>) (valeurs de paramètre autorisées: toutes valeurs)
SVS	choix de l'espacement ligne (<i>select line spacing</i>) (valeurs de paramètre autorisées: toutes valeurs)
SLS	établissement de l'espacement ligne (<i>set line spacing</i>) (valeurs de paramètre autorisées: toutes valeurs)
SGR	établissement de la reproduction graphique (<i>select graphic rendition</i>) (valeurs de paramètre autorisées: 0, 1, 2, 3, 4, 9 à 19, 21 à 24, 29)
STAB	tabulation sélective (<i>selective tabulation</i>) (valeurs de paramètre autorisées: toutes valeurs)
SRS	début de chaîne inverse (<i>start reverse string</i>) (paramètres autorisés: tous paramètres)
VPB	position de ligne en arrière (<i>line position backward</i>) (paramètres autorisés: tous paramètres)
VPR	position de ligne en avant (<i>line position relative</i>) (paramètres autorisés: tous paramètres)
PLD	interligne partiel vers le bas (<i>partial line down</i>)
PLU	interligne partiel vers le haut (<i>partial line up</i>)
BPH	coupure autorisée ici (<i>break permitted here</i>)
NBH	pas de coupure ici (<i>no break here</i>)
JFY	non justifié (<i>no justify</i>)
SUB	caractère de remplacement (<i>substitute</i>)
SP	espace (<i>space</i>)
CR	retour chariot (<i>carriage return</i>)
LF	changement de ligne (<i>line feed</i>)
SOS	début de chaîne (<i>start of string</i>)
ST	fin de chaîne (<i>string terminator</i>)
GCC	composition de caractères graphiques (<i>graphic character composition</i>) (valeurs de paramètre autorisées: toutes valeurs)
IGS	identification de sous-répertoire (<i>identify graphic subrepertoire</i>) (valeurs de paramètre autorisées: toutes valeurs)
HPB	retour-arrière de caractères (<i>character position backward</i>) (valeurs de paramètre autorisées: toutes valeurs)
HPR	avancée relative de caractères (<i>character position relative</i>) (valeurs de paramètre autorisées: toutes valeurs)
SACS	choix de l'espacement des caractères additionnels (<i>set additional character spacing</i>) (valeurs de paramètre: toutes valeurs)

SRCS	choix de l'espacement des caractères réduits (<i>set reduced character spacing</i>) (valeurs de paramètre: toutes valeurs)
SSW	choix de la largeur du caractère ESPACE (<i>set space width</i>) (valeurs de paramètre: toutes valeurs)
PTX	textes parallèles (<i>parallel text</i>) fonctions de commande d'extension de code (voir 6.5.1.4).

L'utilisation de toutes ces fonctions de commande, à l'exception des fonctions SP, CR, LF, SOS et ST, est décrite dans divers paragraphes du 6.5.1.

6.5.1.25 Formatage du contenu

L'attribut «indicateur de formatage» peut être spécifié pour des objets de base particuliers conformes à la présente Recommandation.

D'où la transmission d'objets formatés (ou retraitables formatés) pour lesquels l'emplacement précis de tel ou tel caractère a été entièrement calculé et toutes les fonctions de commande nécessaires incluses. Il en résulte que la plupart des opérations normalement effectuées par le processus associé aux images lors du traitement du texte formaté deviendront inutiles.

Pour ce faire, le processus associé aux images du destinataire doit opérer avec une police de caractères contenant une métrologie identique à celle qu'utilise l'émetteur.

6.5.1.26 Composition des caractères graphiques

Une chaîne de deux ou plus de deux caractères peut être combinée pour former un seul caractère à l'aide de la fonction de commande composition de caractères graphiques (GCC) (*graphic character composition*). Les valeurs de paramètre 0, 1 et 2 peuvent être spécifiées.

6.5.1.27 Textes parallèles

Une chaîne de caractères d'une partie de contenu associée à un composant de base peut être spécifiée comme devant être restituée parallèlement à une autre chaîne de caractères. «ruby» est un exemple type en japonais.

Dans le cas d'un contenu retraitable et d'un contenu formaté retraitable, il est possible de spécifier des textes parallèles à l'aide de la fonction de commande PTX (*parallel text*).

Dans le cas d'un contenu formaté, on peut utiliser les fonctions de commande HPR (*character position relative*), HPB (*character position backward*), VPR (*line position relative*) et VPB (*line position backward*) pour spécifier des textes parallèles. Ces fonctions de commande peuvent aussi être présentes dans un contenu formaté retraitable à condition qu'elles soient contenues dans des chaînes délimitées par les fonctions de commande SOS (*start of string*) et ST (*string terminator*).

6.5.1.28 Crénage couplé

Le crénage couplé peut être spécifié comme devant avoir lieu pendant le processus de formatage, à l'aide de l'attribut «crénage couplé». Ce processus dépend de la police de caractères utilisée et la modification apportée aux positions des caractères dépend de l'information de crénage contenue dans les attributs de polices de caractères. Le crénage couplé ne peut être exécuté que si l'on utilise une police de caractères à espacement variable, l'attribut «crénage couplé» est ignoré si l'on utilise une police de caractères à espacement constant.

6.5.2 Contenu graphique en points

6.5.2.1 Introduction

Le présent paragraphe définit les caractéristiques qui sont applicables au contenu graphique en points renfermé dans un document. Ces caractéristiques peuvent s'appliquer aux composants logiques et physiques de base, sauf indication contraire.

Les valeurs par défaut des caractéristiques ci-après peuvent être spécifiées dans le profil du document:

- type de codage;
- compression;
- espacement des éléments d'image;

- rapport d'espacement;
- dimensions de l'image;
- trajet des éléments d'image;
- progression des lignes.

La spécification, dans un document, d'une caractéristique non essentielle, par un attribut de présentation ou de codage ou une fonction de commande, doit être indiquée dans le profil du document.

6.5.2.2 Architecture de contenu graphique en points

Seule la classe d'architecture de contenu graphique en points de forme retraitsable formatée peut être utilisée dans des documents conformes au présent profil d'application de document. Ce type de contenu peut être utilisé dans des documents de forme retraitsable, de forme formatée et de forme retraitsable formatée.

En cas d'utilisation d'un contenu graphique en points, une seule portion de contenu peut être associée à un objet ou à une classe d'objets.

L'information de contenu d'une portion de contenu peut être absente, cela afin de permettre la représentation et l'échange de documents dans lesquels des parties du contenu peuvent être fournies, par exemple, pendant l'édition suivante.

Par ailleurs, le processus de mise en page du contenu de dimension pondérable ou fixe peut être utilisé en cas de mise en page et de restitution du contenu selon la spécification des attributs de présentation «espacement des éléments d'image» et «dimensions de l'image» décrits aux 6.5.2.6 et 6.5.2.8. Ces deux formes de processus de formatage du contenu peuvent être utilisées dans un même document.

6.5.2.3 Méthodes de codage du contenu graphique en points

Le contenu peut être codé conformément aux méthodes de codage définies dans les Recommandations T.4 et T.6. Dans le cas de la Recommandation T.4, le codage unidimensionnel ou le codage bidimensionnel peuvent être utilisés. Par ailleurs, le schéma de «codage bitmap» défini dans la Rec. T.417 du CCITT | ISO 8613-7 peut être utilisé. Toutes ces formes de codage peuvent être utilisées dans un même document; elles sont toutes des formes de base. Le mode de codage «non comprimé» peut aussi être utilisé mais en tant que caractéristique non essentielle.

En cas d'utilisation de la méthode de codage de la Recommandation T.4 ou de celle de la Recommandation T.6, la relation entre l'ordre des éléments d'image et l'ordre des bits dans les octets du train de données codé doit être telle que le premier élément d'image dans l'ordre des bits soit attribué au bit de plus faible poids d'un octet. Dans le cas du codage bitmap, l'ordre de codage doit être tel que le premier élément d'image soit attribué au bit de plus fort poids d'un octet.

Dans une portion de contenu, il est nécessaire que les attributs de codage «nombre de lignes» et «nombre d'éléments d'image par ligne» soient spécifiés. La valeur de ces attributs doit être un nombre positif; aucune autre restriction n'est imposée en ce qui concerne les valeurs qui peuvent être spécifiées. En conséquence, la présente Recommandation n'impose aucune limitation de la taille des tableaux d'éléments d'image qui peuvent être utilisés.

Le type de méthode de codage utilisé est spécifié par l'attribut «type de codage». L'utilisation de cet attribut n'est pas obligatoire. Par conséquent, si cet attribut n'est pas spécifié pour une portion de contenu particulière et si la classe d'architectures de contenu spécifiée correspond à la classe d'architectures de contenu graphique en points formaté, la méthode de codage par défaut est présumée être celle de la Recommandation T.6.

6.5.2.4 Trajet des éléments d'image et progression des lignes

La direction du trajet des éléments d'image peut être spécifiée comme étant de 0, 90, 180, ou 270 degrés. Celle de la progression des lignes peut être spécifiée comme étant de 90 ou 270 degrés.

Les valeurs de base sont de 0 degré pour le trajet des éléments d'image et de 270 degrés pour la progression des lignes. Toutes les autres valeurs ne sont pas des valeurs de base et leur utilisation doit être indiquée dans le profil du document.

6.5.2.5 Découpage

Sous-région d'un tableau d'éléments d'image représentée par une portion de contenu associée à un composant de base qui peut être définie à l'aide de l'attribut de présentation «découpage». Aucune restriction n'est imposée à l'utilisation de cet attribut.

6.5.2.6 Espacement des éléments d'image

L'espacement des éléments d'image est la distance en BMU entre deux éléments d'image quelconques sur une ligne au stade de la restitution d'un tableau d'éléments d'image. Toute valeur peut être expressément spécifiée à condition que l'espacement entre les éléments d'image ne soit pas inférieur à 1 BMU. La valeur d'espacement des éléments d'image ne doit pas nécessairement correspondre à un nombre entier. En outre, la valeur nulle peut être spécifiée, ce qui indique que le processus de formatage pondérable doit être utilisé.

La spécification de la valeur nulle ou d'espacements de 16, 12, 8, 6, 5, 4, 3, 2 et 1 BMU entre éléments d'image adjacents correspond aux valeurs de base. La spécification de tout autre espacement est non essentielle et doit être indiquée dans le profil du document.

L'espacement des éléments d'image applicable au contenu associé aux composants logiques de base est spécifié par l'attribut de présentation «espacement des éléments d'image».

NOTES

1 Les valeurs de base de l'attribut «espacement des éléments d'image» énumérées ci-dessus équivalent respectivement à des résolutions de 75, 100, 150, 200, 240, 300, 400, 600 et 1200 éléments d'image par 25,4 mm, pour une BMU de 1/1200 pouces.

2 L'attribut «espacement des éléments d'image» spécifie deux nombres entiers, dont le rapport détermine l'espacement des éléments d'image. Les valeurs de ces nombres entiers ne font l'objet d'aucune restriction.

6.5.2.7 Rapport d'espacement

Le rapport d'espacement est le rapport entre l'espacement des éléments d'image et l'espacement des lignes au stade de la restitution d'un tableau d'éléments d'image. Ce rapport est utilisé pour déterminer l'espacement ligne d'après l'espacement des éléments d'image spécifié.

Il n'est pas imposé de restrictions à la valeur de ce rapport, pour autant que la valeur d'espacement ligne obtenue ne soit pas inférieure à 1 BMU. En outre, la valeur d'espacement ligne ne doit pas nécessairement correspondre à un nombre entier de BMU. Les valeurs sont toutes des valeurs de base.

La valeur par défaut peut être spécifiée dans le profil du document. Si aucune valeur par défaut n'est expressément spécifiée, la valeur par défaut correspond au rapport 1:1, c'est-à-dire que l'espacement ligne est égal à l'espacement des éléments d'image.

Le rapport d'espacement applicable au contenu associé à un composant logique de base est spécifié par l'attribut de présentation «rapport d'espacement».

6.5.2.8 Dimensions de l'image

Les dimensions de l'image sont les contraintes à appliquer à la taille de l'image produite au stade du formatage d'un tableau d'éléments d'image représenté par une portion de contenu associée à un composant logique de base.

Ces contraintes sont spécifiées pour les composants logiques de base par l'attribut de présentation «dimensions de l'image». La valeur de cet attribut n'est prise en compte que si la valeur de l'attribut «espacement des éléments d'image» est la valeur nulle.

6.5.3 Contenu graphique géométrique

Un document peut contenir des images graphiques composées de commandes graphiques géométriques codées sous forme de métafichiers de commandes graphiques (CGM) conformes à la Norme ISO 8632. Chaque figure CGM doit consister en une image unique. Chaque figure CGM peut spécifier les dimensions minimales de cette image.

On trouvera dans l'Annexe B de plus amples informations concernant les spécifications du contenu graphique géométrique.

6.6 Caractéristiques diverses

6.6.1 Documents ressources

Des classes d'objet des types BodyText, BodyRaster, BodyGeometric, CommonText, CommonRaster, CommonGeometric, EntryText, EntryRaster, EntryGeometric et GenericBlock peuvent se rapporter aux contraintes de constitution correspondantes d'un document ressource.

Les contraintes de constitution du document ressource peuvent faire référence à des portions de contenu et des styles de formatage et de présentation que contient le document ressource. Ces contraintes énumérées ci-dessus sont les seules auxquelles il est possible de faire référence depuis un autre document à l'aide de l'attribut ressource. Toutefois les documents génériques utilisés comme documents ressources peuvent contenir une combinaison quelconque de contraintes de constitution génériques conforme à ce profil d'application de document.

6.6.2 Documents externes

Un document externe retraitsable ou retraitsable formaté peut contenir la structure logique générique et la structure physique générique si elle existe. Il faut noter qu'il n'est pas permis de changer une structure générique dans le document échangé tout en faisant référence à l'autre par l'intermédiaire du document externe.

6.6.3 Frontières

Des frontières peuvent être spécifiées pour tous les types de cadres définis aux 6.3.5, 6.3.6 et 6.3.7 à l'aide de l'attribut «frontières». Toutes les caractéristiques des frontières spécifiées dans la Rec. T.412 du CCITT | ISO 8613-2, peuvent être spécifiées comme étant des valeurs de base. Des frontières peuvent également être spécifiées dans des styles de présentation.

6.6.4 Commentaires d'application

La spécification de l'attribut «commentaires d'application» est obligatoire pour toutes les classes d'objets contenues dans un document conforme à la présente Recommandation. La spécification et l'utilisation de cet attribut sont facultatives.

Cet attribut est structuré de manière qu'il contienne deux champs. Le premier champ est obligatoire lorsque l'attribut est spécifié et contient une chaîne numérique qui identifie de manière unique la contrainte de constitution applicable au constituant pour lequel l'attribut est spécifié. Cela facilite le traitement des documents. Une liste de ces identificateurs est donnée dans les Tableaux 5 et 6.

NOTES

1 Les valeurs des identificateurs numériques de contraintes de constitution ne sont pas les mêmes pour une structure logique ou physique; par conséquent, pour identifier la contrainte de constitution applicable à un constituant, il faut connaître la structure dont fait partie ce constituant.

2 Pour les contraintes de constitution qui se correspondent entre profils hiérarchiquement apparentés auxquels appartient la présente Recommandation, le même identificateur numérique de contrainte sur les constituants est spécifié.

Le deuxième champ est facultatif et peut contenir toute information relative à l'application ou à l'utilisateur. Le format du deuxième champ n'est pas défini dans la présente Recommandation et l'interprétation de ce champ dépend d'un accord privé entre l'expéditeur et le destinataire du document.

Le codage de l'attribut «commentaires d'application» est défini aux 8.1.3. et 8.2.3.

6.6.5 Représentation de remplacement

L'information qui figure dans une portion de contenu peut être remplacée par une chaîne de caractères spécifiée dans l'attribut «représentation de remplacement». Cet attribut peut être spécifié dans les portions de contenu qui renferment un contenu caractères, un contenu graphique en points ou un contenu graphique géométrique.

La spécification et l'utilisation de cet attribut sont facultatives. La chaîne de caractères spécifiée doit appartenir aux répertoires de caractères indiqués dans l'attribut de profil de document «jeux de caractères de représentation de repli» (voir 6.7.4.3). Si ce dernier attribut n'est pas expressément spécifié dans le profil du document, le sous-répertoire minimal défini dans les Rec. de la série T.410 du CCITT | ISO 8613 constitue alors le jeu de caractères par défaut. Les fonctions de commande SP (*space*), CR (*carriage return*) et LF (*line feed*) peuvent aussi être utilisées dans la chaîne de caractères mais aucune autre fonction de commande n'est autorisée; les jeux de caractères graphiques ne peuvent donc être modifiés en aucun point de la représentation de remplacement.

6.6.6 Mécanismes de numérotation automatique et de référence

6.6.6.1 Introduction

Cette Recommandation fournit des mécanismes généraux de numérotation automatique des divers types de constituants d'un document, ainsi que de référence aux numéros ainsi affectés à partir d'autres constituants du document. Est ainsi assurée, par exemple, la numérotation de segments (tels que des chapitres, paragraphes et annexes) de tableaux, figures, notes de bas de page, listes d'éléments et pages.

Des chaînes de caractères peuvent en outre être spécifiées pour les divers constituants d'un document, et il peut être fait référence à ces chaînes à partir d'autres parties du document, pour assurer un mécanisme général de référence à l'intérieur de ce document.

Pour réaliser ces fonctions, cette Recommandation assure les affectations indiquées au 6.6.6.2. Les paragraphes 6.6.6.3 à 6.6.6.11 décrivent l'utilisation de ces affectations pour la numérotation automatique et les schémas de référence. Ces descriptions n'ont pas pour objet de restreindre l'utilisation des affectations assurées par cette Recommandation, ni des mécanismes utilisés pour réaliser cette numérotation et ces schémas de référence.

TABLEAU 5/T.506

**Liste des identificateurs de chaînes numériques pour les
contraintes de constitution logiques**

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

TABLEAU 6/T.506

**Liste des identificateurs de chaînes numériques de contraintes
de constitution physiques**

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

6.6.6.2 Affectations

Les affectations spécifiées dans les paragraphes suivants peuvent être attribuées, sauf spécifications différentes, à tout constituant logique composite et à tout constituant logique de base des types BodyText, BodyRaster et BodyGeometric, ainsi qu'aux constituants physiques de types DocumentLayoutRoot, PageSet, Page, RectoPage et VersoPage.

Les groupes d'affectations ont des noms de forme générale '<name>-<n>'. La chaîne de caractères <name> sert à identifier un groupe particulier d'affectations, et <n> est une chaîne de caractères qui identifie une affectation particulière. Le champ <n> est une séquence de caractères pris dans l'ensemble de caractères '0...9'; cette séquence peut être de n'importe quelle longueur mais doit être composée d'une chaîne représentant un nombre entier sans zéro non significatif.

Les valeurs des affectations peuvent consister en entiers ou en chaînes de caractères. Dans le cas d'entiers, toute valeur peut être spécifiée. Dans le cas de chaîne de caractères, la chaîne peut comprendre n'importe lesquels des 94 caractères de l'IRV d'ISO 646 (révisée 1991) plus le caractère espace. Tout autre répertoire de caractères peut être utilisé, à condition qu'il soit désigné et appelé par les annonceurs d'extension de code appropriés et indiqué dans le profil du document comme une valeur qui n'est pas de base. Aucune autre fonction de commande ne peut être utilisée.

6.6.6.2.1 Affectation 'prefixes-<n>'

Cette affectation spécifie une chaîne de caractères qui est en général utilisée pour spécifier une chaîne préfixe dans une chaîne de caractères représentée par une autre affectation. Exemples: préfixes de numéro de segment, de note de bas de page et de page.

6.6.6.2.2 Affectation 'suffixes-<n>'

Cette affectation spécifie une chaîne de caractères qui est en général utilisée pour spécifier une chaîne suffixe dans une chaîne de caractères représentée par une autre affectation. Exemples: suffixes de numéro de segment, de note de bas de page et de page.

6.6.6.2.3 Affectation 'numberstring-<n>'

Cette affectation spécifie une chaîne de caractères qui comprend en général un ou plusieurs nombres et séparateurs constituant, par exemple, un numéro de segment, figure, tableau, élément de liste ou note de bas de page, dans un document. Exemple: la chaîne «3.4.3.6» qui pourrait identifier une sous-section de document.

6.6.6.2.4 Affectation 'numbers-<n>'

Cette affectation spécifie un entier qui est associé à un constituant particulier. Cet entier est en général utilisé pour générer un chiffre ou une séquence de chiffres représenté par l'affectation 'numberstring-<n>', qui identifie, par exemple, un segment, une note de bas de page, un tableau, une figure, un élément de liste ou une page dans un document.

6.6.6.2.5 Affectation 'separator-<n>'

Cette affectation spécifie une chaîne de caractères qui est typiquement utilisée pour représenter des séparateurs entre les chiffres d'une chaîne, représentée par l'affectation 'numberstring-<n>'. Exemple: la chaîne «3.4.3.6» où le caractère «.» forme le séparateur.

6.6.6.2.6 Affectation 'string-<n>'

Cette affectation spécifie une chaîne de caractères qui est associée à un constituant et utilisée pour assurer un mécanisme de référence général associé à un document. Généralement, cette affectation est utilisée pour assurer la référence du contenu caractères spécifié dans une partie du document, à partir d'une autre partie du document. Exemple: cette association peut être utilisée pour véhiculer le titre d'un chapitre ou d'une figure qui est référencé dans une autre partie du document.

6.6.6.2.7 Affectation 'PGnum'

Cette affectation spécifie un entier qui représente généralement un numéro de page. Elle peut être uniquement spécifiée pour des contraintes de constitution physiques des types DocumentLayoutRoot, PageSet, RectoPage, VersoPage et Page.

6.6.6.2.8 Affectation 'fnotenumber'

Cette affectation spécifie un entier qui est spécifiquement fourni pour présenter les numéros qui identifient les notes de bas de page. Cette affectation peut uniquement être spécifiée pour les contraintes de constitution logiques DocumentLogicalRoot, Passage et Footnote et est fournie pour assurer la compatibilité avec FOD 026 | PM-26.

6.6.6.2.9 Affectation 'fnotestring'

Cette affectation spécifie une chaîne de caractères qui est l'équivalent du numéro représenté par l'affectation 'fnotenumber'. Elle ne peut être spécifiée que pour la contrainte de constitution logique Footnote et est fournie pour compatibilité avec FOD 026 | PM-26.

6.6.6.3 Numérotation de segments

Le constituant Number contient un générateur de contenu qui, sur interrogation d'évaluation durant le processus de formatage, produit une chaîne de caractères qui sert à identifier le NumberedSegment auquel ce constituant Number appartient.

Le format de cette chaîne de caractères est le suivant:

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

Ce format est défini par une expression de chaîne qui est spécifiée par la macro SEGMENTNUMBER (voir 7.3.1). La description ci-après indique comment cette chaîne est typiquement générée.

Les champs `<pre-str>` et `<suf-str>` sont des chaînes de caractères de préfixe et de suffixe qui peuvent être de n'importe quelle longueur. Elles peuvent être prédéfinies dans l'expression ou dérivées d'affectations de types 'prefixes-`<n>`' et 'suffixes-`<n>`', qui sont définies sur des constituants de plus haut niveau de la structure du document.

Le champ `<num-str>` est l'identificateur de segment qui a la forme générale suivante:

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

où [...]... indique une répétition optionnelle. Autrement dit, un identificateur de segment peut consister en un seul chiffre ou en une séquence de deux chiffres ou plus, tous séparés par une chaîne de caractères appelée 'séparateur'. Exemple: la chaîne «7.3.3.1».

Les identificateurs de segment sont représentés par l'affectation 'numberstring-`<n>`'. Cette affectation peut être explicitement spécifiée ou générée automatiquement par une expression de type chaîne définie par la macro USENUMBERSTRINGS (voir 7.3.1) qui a la forme générale:

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

Le champ `<numberstring-x>` est une référence à l'identificateur de segment (c'est-à-dire une autre instance d'affectation du type '`<numberstring>-<n>`' qui est spécifiée pour le NumberedSegment immédiatement supérieur. Ainsi est autorisée la spécification de schéma de numérotation à structuration hiérarchique. Si ce NumberedSegment n'existe pas, ce champ est vide, et dans ce cas, non hiérarchique.

Le champ `<separator-y>` est une chaîne de caractères dérivée d'une association de type 'separator-`<n>`' spécifiée à un niveau plus élevé de la structure du document. Ce champ peut être vide.

Le champ `<number-z>` est le numéro, représenté sous la forme d'une chaîne de caractères, qui est applicable au NumberedSegment dont l'identificateur est en cours de construction. Ce numéro peut être représenté sous la forme d'une chaîne numérique de chiffres arabes, d'une chaîne numérique de chiffres romains majuscules ou minuscules, ou par des caractères alphabétiques majuscules ou minuscules.

La valeur entière correspondant au champ `<number-z>` peut être générée par une des deux méthodes suivantes: la valeur peut être générée par une fonction ORDINAL à l'intérieur de l'expression définie par la macro USENUMBERSTRINGS. Elle peut également être générée par une affectation du type 'number-`<n>`' qui est spécifiée pour le NumberedSegment dont l'identificateur est en cours de construction. L'affectation 'numbered-`<n>`' est initialisée en un point approprié du document, et est ensuite incrémentée automatiquement sur chaque NumberedSegment successif. Cette incrémentation est réalisée en utilisant une expression définie par la macro USENUMBERS (voir 7.3.1).

Il peut alors être fait référence à l'affectation ainsi construite 'numberstring-`<n>`' générée par la macro USENUMBERSTRINGS par un générateur de contenu spécifié par le constituant Number pour produire l'identificateur du NumberedSegment comme indiqué plus haut dans ce paragraphe.

Cette affectation est également disponible pour construire les identificateurs de segments utilisés à des niveaux inférieurs du NumberedSegment. Autrement dit, un schéma de numérotation hiérarchique peut être spécifié pour des NumberedSegments à différents niveaux de la structure du document.

Le numéro du niveau doit être indiqué à l'aide du plus petit nombre possible de caractères; en d'autres termes, il ne doit pas y avoir de zéro non significatif.

Un document peut contenir n'importe quel nombre de schémas de numérotation indépendants différents. Ils sont spécifiés en positionnant les valeurs des affectations des types 'numbers-<n>', 'numberstring-<n>', 'prefixes-<n>' et 'suffixes-<n>', ainsi que les expressions indiquées ci-dessus, aux points appropriés de la structure du document.

Les mécanismes ci-dessus peuvent être utilisés à différentes fins; les paragraphes suivants indiquent l'utilisation typique de ce mécanisme pour la numérotation de figures, tableaux, listes d'éléments et notes de bas de page.

6.6.6.4 Numérotation des figures

Les mécanismes utilisés pour la numérotation des figures est le même que celui appliqué pour la numérotation des segments (voir 6.6.6.3). Autrement dit, le numéro d'une figure est produit par un générateur de contenu spécifié dans le constituant Number qui est immédiatement subordonné à la figure donnée. Ce générateur de contenu contient une expression dont la valeur est définie par la macro SEGMENTNUMBERS. Le numéro associé à chaque figure peut être représenté par une affectation de type 'number-<n>' et une association 'numberstring-<n>' est utilisée pour spécifier la chaîne de caractères qui représente le numéro de figure. La génération de ces affectations est décrite au 6.6.6.3.

Les figures d'un document peuvent être numérotées consécutivement dans tous les documents, quelle que soit la partie de ce document à laquelle ces figures appartiennent. Une autre possibilité est d'affecter les numéros à des parties du document auxquelles les figures appartiennent; par exemple, les figures du Chapitre 3 d'un document peuvent être spécifiées comme 3.1, 3.2, 3.3, etc.

6.6.6.5 Numérotation de listes

Une liste d'éléments individuellement numérotés est représentée par le constituant logique NumberedList. Chaque élément est représenté par un constituant de type ListItem et le numéro associé à chaque élément est représenté par un constituant précédent du type Number.

Number contient un générateur de contenu qui, sur interrogation d'évaluation, produit le numéro associé à l'élément suivant. Le format de ce générateur de contenu est le même que celui utilisé pour la numérotation des segments (voir 6.6.6.3) et est défini par la macro SEGMENTNUMBERS.

Comme décrit au 6.6.6.3, l'affectation 'numberstring-<n>' est utilisée pour représenter la chaîne numérique associée à chaque élément. Dans ce cas, l'affectation est générée par le constituant Number auquel se réfère le générateur de contenu contenu dans le même constituant. (A noter que dans le cas de numéro de segment, l'affectation 'numberstring <n>' est générée sur l'objet supérieur.)

6.6.6.6 Numérotation de tableaux

La contrainte de constitution TableNumber est en général utilisée pour représenter une chaîne de caractères qui constitue le numéro associé à un Table. Cette chaîne est formatée dans le cadre d'en-tête au moyen d'un cadre SourcedContentFixed. Ce dernier cadre spécifie l'attribut «source logique» qui indique l'instance du constituant CommonContent qui contient le subordonné TableNumber qui spécifie le numéro de tableau requis.

La chaîne de caractères représentée par TableNumber est produite par un générateur de contenu qui définit une expression chaîne spécifiée par la macro TABLENUMBERS (voir 7.3.1). Le format général de cette chaîne de caractères est le suivant:

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

Les champs <pre-str> et <suf-str> sont des chaînes de caractères optionnelles préfixe et suffixe, qui sont prédéfinies dans l'expression ou dérivées d'affectations de types 'prefixes-<n>' et 'suffixes-<n>', qui sont spécifiées pour des constituants de niveau supérieur de la structure du document.

Le champ <num-str> est une chaîne de caractères qui représente l'identificateur du tableau en cours de formatage. Il est obtenu à partir de l'affectation 'numberstring-<n>' qui est spécifiée pour l'objet logique du type Table qui est en cours de formatage. Autrement dit, le champ <num-str> dérive de l'instance courante du constituant du type Table.

Le paragraphe 6.6.6.3 décrit le format général du champ <num-str> et les mécanismes utilisés pour spécifier et générer ce champ.

Les tableaux, comme les figures, peuvent être numérotés indépendamment dans l'ensemble d'un document, ou avoir des numéros affectés aux segments qui les contiennent. Un exemple de numéro de tableau est la chaîne «1.1.5» où «1.1» est le numéro du segment auquel le tableau appartient. «.» est un séparateur et «5» est le numéro associé au tableau.

6.6.6.7 Numérotation de notes de bas de page

Les contraintes de constitution FootnoteReference et FootnoteNumber contiennent des générateurs de contenu qui, sur interrogation d'évaluation durant le processus de formatage, produisent des chaînes de caractères qui servent à identifier la Footnote à laquelle les constituants FootnoteReference et FootnoteNumber sont subordonnés.

Le format de cette chaîne de caractères est le suivant:

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

Cette chaîne est définie par l'expression chaîne spécifiée par la macro FNNUMBER (voir 7.3.1).

Le mécanisme de numérotation est le même que celui utilisé pour la numérotation des segments (voir 6.6.6.3). Ainsi, les champs <pre-str> et <suf-str> sont des chaînes de caractères préfixe et suffixe optionnelles, de n'importe quelle longueur. Elles dérivent d'affectations de types 'prefixes-<n>' et 'suffixes-<n>' qui sont définies sur des constituants de niveau supérieur de la structure du document.

Le champ <num-str> peut être dérivé par l'une des trois méthodes suivantes: il peut être représenté par des affectations du type 'numberstring-<n>' qui sont dérivées en utilisant une expression spécifiée par la macro USENUMBERSTRINGS comme décrit au 6.6.6.3. Une seconde méthode est de le dériver de l'affectation du type 'fnotestring' qui est spécifiée sur le constituant Footnote auquel les contraintes de constitution FootnoteReference et FootnoteNumber sont subordonnées. Ce champ est automatiquement généré en utilisant des expressions définies par les macros INCFNOTENUMBER et FNOTENUMBERSTRING. Enfin, le champ <num-str> peut être spécifié explicitement; ce cas est défini par la macro FNOTESTRINGLITERAL.

Les mécanismes ci-dessus permettent la numérotation consécutive de notes de bas de page dans l'ensemble d'un document ou l'utilisation de n'importe quel nombre de schémas de numérotation de notes de bas de page indépendants. Par exemple, les notes de bas de page applicables à des segments, des figures et des tableaux peuvent être numérotées indépendamment.

6.6.6.8 Numérotation des pages

La contrainte de constitution PageNumber est fournie spécifiquement pour représenter le contenu commun qui contient un numéro de page et qui doit être placé sur les pages successives d'un document. Un mécanisme assure l'incrémement automatique des numéros des pages successives d'un document.

Le format du générateur de contenu spécifié par le constituant PageNumber est le suivant:

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

Ce format est défini par une expression chaîne spécifiée par la macro PGNUMBERS (voir 7.3.1).

Les champs <pre-str> et <suf-str> sont des chaînes de caractères préfixe et suffixe optionnelles qui peuvent être de n'importe quelle longueur. Elles peuvent être explicitement spécifiées dans l'expression ou être dérivées d'affectations de types respectifs 'prefixes-<n>' et 'suffixes-<n>' qui sont définies sur des constituants de niveau supérieur dans la structure du document.

Le champ <num-str> est le numéro de page. Il consiste en un numéro unique dérivé de l'affectation 'PGnum' qui est spécifiée pour l'instance courante du cadre ou de la page auquel le numéro de page doit être affecté. Un numéro de page peut être représenté sous la forme de chaînes numériques de chiffres arabes, de chaînes numériques de chiffres romains majuscules ou minuscules ou de chaînes alphabétiques équivalentes majuscules ou minuscules.

L'affectation 'number-<n>' est initialisée au niveau de la racine physique du document, de l'ensemble de pages ou d'une classe de pages particulière (à l'aide de la macro INITIALISEBINDINGS définie au 7.3.1). L'affectation PGnum est initialisée au niveau de la racine physique du document ou de l'ensemble de pages (en utilisant la macro INITIALISEPGNUM définie au 7.3.1). Cette affectation est incrémentée automatiquement à chaque page successive, en utilisant l'expression spécifiée par la macro USEPGNUMBERS (voir 7.3.1). En plaçant l'initialisation sur la racine physique et non sur des classes d'ensembles de pages, la numérotation des pages peut être continue d'un ensemble de pages au suivant.

Le contenu associé aux classes d'objets logiques du type PageNumber est formaté dans un cadre de l'un des types suivants: BasicHeader, BasicFooter, SourcedContentVariable, SourcedContentFixed (voir 6.3.6), en utilisant le

mécanisme de source logique. Ainsi, lors du formatage du cadre approprié, le champ <num-expr> du générateur de contenu contenu dans la classe d'objets logiques du type PageNumber est évalué, ce qui détermine la valeur de l'affectation PGnum qui est associée à la page courante en cours de formatage.

Une numérotation similaire s'applique à des ensembles de pages.

6.6.6.9 Contenu référencé

ReferencedContent est une contrainte de constitution qui est fournie pour assurer un mécanisme de référence général dans le contenu d'une partie de corps d'un document. Ce constituant est donc utilisé pour représenter un contenu caractères qui contient une référence à un contenu spécifié ailleurs dans un document. Exemple: références à des chaînes qui représentent des numéros de segments, de figures, de tableaux, de notes de bas de page et de pages. Il est possible de faire référence à des chaînes particulières spécifiées par 'string-<n>'.

Cette contrainte de constitution contient un générateur de contenu qui, sur interrogation d'évaluation, produit une chaîne de caractères de la forme suivante:

<pre-str><ref-str><suf-str>

Ce générateur de contenu est défini par une expression chaîne qui est spécifiée par la macro REF (voir 7.3.1).

Les champs <pre-str> et <suf-str> sont des chaînes de caractères préfixes et suffixes optionnelles qui peuvent être définies explicitement dans l'expression ou être dérivées d'affectations de types respectifs 'prefixes-<n>' et 'suffixes-<n>', qui sont définies sur des constituants de niveau supérieur dans la structure du document.

Le champ <ref-str> est une chaîne de caractères qui est obtenue à partir du contenu spécifié dans un constituant particulier du document par référence à une des affectations suivantes: 'numberstring-<n>', 'numbers-<n>', 'strings-<n>', 'fnotestring' ou 'PGnum'.

Dans ce cas, les mécanismes de référence suivants sont autorisés:

- référence à un objet logique particulier pour l'affectation requise;
- référence à un objet logique particulier et recherche de l'affectation requise sur les objets logiques supérieurs à l'objet référencé.

Le mécanisme de référence doit tenir compte des constituants pour lesquels des affectations particulières sont autorisées, tels que définies au 6.6.6.2.

6.6.6.10 Références communes

CommonReference est une contrainte de constitution appartenant à la partie commune de la structure logique d'un document, qui représente un contenu caractères commun qui peut être reproduit en plusieurs endroits de ce document. Cette contrainte est fournie spécifiquement pour représenter un contenu qui contient une référence à un contenu spécifié ailleurs dans un document. Ce contenu est spécifié par des affectations. Exemple: références à des chaînes de caractères qui représentent des numéros de segments, de figures, de tableaux et de notes de bas de page d'un document.

Cette contrainte de constitution contient un générateur de contenu, dont le format général est le même que celui du générateur de contenu spécifié pour le constituant ReferencedContent (voir 6.6.7.9) et dont le contenu peut être obtenu à partir des affectations suivantes: 'numberstring-<n>', 'numbers-<n>', 'strings-<n>', 'fnotestring' ou 'PGnum'.

Ce générateur de contenu est défini par une expression qui est spécifiée par la macro COMMONREF.

Dans ce cas, les mécanismes de référence suivants sont autorisés:

- référence à une page ou un cadre courant ou un objet logique, qui est présenté pour l'affectation requise;
- référence à une page ou un cadre courant qui est présenté; une recherche de l'affectation requise est effectuée sur les constituants supérieurs à un constituant référencé;
- référence à une page ou un cadre particulier dans laquelle/lequel un objet logique est présenté pour l'affectation requise;
- référence à une page ou un cadre dans laquelle/lequel un objet logique particulier est présenté; une recherche de l'affectation requise est faite sur les objets physiques supérieurs à la page/au cadre référencé(e).

Le mécanisme de référence doit tenir compte des constituants pour lesquels les affectations particulières sont autorisées (6.6.6.2).

6.6.6.11 Références à l'instance courante

La contrainte de constitution `CurrentInstance`, comme `CommonReference`, représente un contenu caractères commun qui peut être reproduit en plusieurs endroits d'un document lors de son formatage.

Cette contrainte est fournie spécifiquement pour représenter l'instance courante d'une chaîne de caractères qui doit être disposée en plusieurs endroits d'un document. Un exemple typique est la reproduction d'un titre de chapitre à chaque page d'un document dans lequel ce chapitre est reproduit.

La contrainte de constitution `CurrentInstance` contient un générateur de contenu, dont le format général est le même que celui du générateur de contenu spécifié pour la contrainte de constitution `ReferencedContent`. Ce générateur de contenu est restreint au contenu référencé obtenu à partir des affectations de type `'string-<n>'`.

Le mécanisme de référence permet la référence à l'affectation du type `'string-<n>'` pour l'instance courante d'un constituant logique d'une classe spécifiée ou pour l'instance courante d'un cadre particulier ou d'une page particulière.

Le format de ce générateur de contenu est défini par la macro `CURRENTINSTANCE` (voir 7.3.1).

6.6.6.12 Identificateurs communs

La contrainte de constitution `CommonNumber` représente un contenu caractères commun qui peut être reproduit en plusieurs endroits d'un document lors de son formatage.

Cette contrainte est fournie spécifiquement pour représenter des identificateurs d'autres parties d'un document, c'est-à-dire des chaînes de caractères qui représentent les numéros de segments, figures et tableaux d'un document.

Cette contrainte contient un générateur de contenu, dont le format général est le même que celui du générateur de contenu spécifié pour le constituant `ReferencedContent` (voir 6.6.6.9). Ce générateur de contenu peut référencer des contenus spécifiés par les affectations `'numberstring-<n>'` et `'numbers-<n>'`.

Le format de ce générateur de contenu est défini par la macro `COMMONNUMBER` (voir 7.3.1).

6.6.7 Commentaires lisibles par l'utilisateur

Information à interpréter comme des commentaires concernant des constituants et des portions de contenu associées et pouvant être spécifiée à l'aide de l'attribut «commentaires lisibles par l'utilisateur». Cette information est destinée à être présentée à des personnes.

L'information consiste en une chaîne de caractères qui doit faire partie de l'un des répertoires de caractères indiqués dans l'attribut du profil du document «jeux de caractères de commentaires» (voir 6.7.4.2). Si ce dernier attribut n'est pas expressément spécifié, le jeu de caractères par défaut est alors le sous-répertoire minimum de Recommandations de la série T.410 du CCITT | ISO 8613. Les fonctions de commande SP, CR, LF et les fonctions de commande d'extension de code peuvent aussi être utilisées en tout point de la chaîne de caractères, mais il n'est pas admis d'autres fonctions de commande.

6.6.8 Nom visible par l'utilisateur

Information utilisable pour identifier les constituants d'un document et pouvant être spécifiée à l'aide de l'attribut «nom visible par l'utilisateur». Cette information est destinée à être présentée à des personnes, par exemple à titre d'aide à l'édition de documents.

L'information consiste en une chaîne de caractères qui doit faire partie de l'un des répertoires de caractères indiqués dans l'attribut du profil du document «jeux de caractères de commentaires» (voir 6.7.4.2). Si ce dernier attribut n'est pas expressément spécifié, le jeu de caractères par défaut est alors le sous-répertoire minimum de Recommandations de la série T.410 du CCITT | ISO 8613. Les fonctions de commande SP, CR, LF et les fonctions de commande d'extension de code peuvent aussi être utilisées en tout point de la chaîne de caractères, mais il n'est pas admis d'autres fonctions de commande.

6.6.9 Unit scaling

L'attribut de profil de document `'unit scaling'` peut être spécifié pour indiquer un facteur d'échelle qui doit être appliqué à tous les attributs et aux valeurs des fonctions de commande qui spécifient des positions ou des dimensions absolues ou relatives. Cet attribut spécifie deux nombres entiers `m` et `n` qui indiquent que ces valeurs doivent être interprétées comme étant égales à `m/n MU`.

6.7 Caractéristiques de gestion du document

Les informations relatives au document dans son ensemble sont spécifiées dans le profil du document qui est représenté par le constituant *DocumentProfile*, lequel doit être spécifié dans chaque document.

Les informations contenues dans le profil du document sont classées dans les catégories suivantes:

- information sur les constituants du document;
- information d'identification du document;
- information sur les valeurs par défaut du document;
- information sur les caractéristiques non essentielles;
- information sur la gestion du document.

Les informations contenues dans le profil du document peuvent présenter de l'intérêt pour l'utilisateur ou peuvent être utilisées pour le traitement du document par une machine.

6.7.1 Information sur les constituants du document

Cette information spécifie les constituants utilisés pour représenter le document, y compris les constituants externes au document échangé. Cette information est divisée en trois catégories.

6.7.1.1 Présence des constituants du document

Cette information indique les constituants inclus dans le document. En d'autres termes, cette information indique si le document contient ou non une structure logique générique, une structure logique spécifique, une structure physique générique, une structure physique spécifique, des styles de formatage et des styles de présentation (voir la Note). Il est obligatoire de spécifier cette information dans le profil du document.

NOTE – Si la structure logique (ou physique) générique est externe au document (voir 6.7.1.3), il demeure nécessaire d'indiquer que ces structures sont présentes et qu'elles font partie du document.

6.7.1.2 Information sur les documents ressources

Cette information consiste en un renvoi à un document ressource (voir 6.6.1). Cette information est spécifiée par l'attribut «document ressource». Si des constituants du document font référence à des classes d'objets d'un document ressource, il est obligatoire en ce cas de spécifier cette information dans le profil du document.

6.7.1.3 Information sur les documents externes

Cette information consiste en un renvoi à un document externe qui peut consister en une structure logique générique ou en une structure physique générique ou encore en l'une et l'autre de ces structures (voir 6.6.2). Si un tel renvoi est nécessaire, il est spécifié en ce cas par l'attribut «classe de document externe» dans le profil du document.

6.7.2 Information d'identification du document

Cette information se rapporte à l'identification du document. Elle est divisée en six catégories.

6.7.2.1 Information sur le profil d'application de document

Cette information indique le profil d'application dont le document relève. Il est obligatoire de spécifier cette information à l'aide de l'attribut «profil d'application de document».

6.7.2.2 Information sur la classe d'architecture de document

Cette information indique la classe d'architecture à laquelle le document appartient (voir 6.1). Il est obligatoire de spécifier cette information à l'aide de l'attribut «classe d'architecture de document».

6.7.2.3 Information sur la classe d'architecture de contenu

Cette information indique les classes d'architectures de contenu utilisées dans le document (voir 6.5.1.2, 6.5.2.2 et 6.5.3). Il est obligatoire de spécifier cette information à l'aide de l'attribut «classe d'architecture de contenu».

6.7.2.4 Information sur la classe de format d'échange

Cette information indique la classe de format d'échange utilisée pour présenter le document (voir l'article 8). Il est obligatoire de spécifier cette information à l'aide de l'attribut «classe de format d'échange».

6.7.2.5 Information sur la version ODA

Cette information indique la Norme de l'ISO ou la Recommandation du CCITT à laquelle le document est conforme. En outre, elle spécifie une date, qui indique que le document est conforme à la version de la Norme de l'ISO ou de la Recommandation du CCITT et les éventuels addenda en vigueur à cette date. Il est obligatoire de spécifier cette information à l'aide de l'attribut «version ODA».

6.7.2.6 Référence du document

Cette information sert à identifier le document. En règle générale, cette information est attribuée au document par l'auteur. L'identificateur peut consister en un identificateur d'objet ASN.1 ou en une chaîne de caractères. Il est obligatoire de spécifier cette information à l'aide de l'attribut «référence du document».

6.7.3 Information sur les valeurs par défaut du document

Cette information spécifie diverses valeurs par défaut pour les attributs utilisés dans le document. Les valeurs par défaut qui sont autorisées sont spécifiées dans les divers paragraphes de l'article 6 du présent profil. Cette information n'est obligatoire que lorsqu'il est nécessaire de spécifier une valeur par défaut autre que la valeur par défaut normalisée qui est définie dans les Rec. de la série T.410 du CCITT | ISO 8613.

Des valeurs par défaut peuvent être spécifiées pour les groupes d'attributs suivants:

- attributs d'architecture de document;
- attributs de contenu en caractères;
- attributs de contenu graphique en points;
- attributs de contenu graphique géométrique.

6.7.4 Information sur les caractéristiques non essentielles

Cette information précise les valeurs d'attribut non essentielles spécifiées dans le document. Il est obligatoire d'indiquer un attribut non essentiel dans le profil du document lorsqu'une telle valeur est utilisée dans le document.

Les types suivants d'attribut non essentiel peuvent être spécifiés:

- jeux de caractères de profil de document;
- jeux de caractères de commentaires;
- jeux de caractères de représentation de repli;
- dimensions de pages;
- type de support;
- caractéristiques de présentation des caractères;
- caractéristiques de présentation de contenu graphique en points;
- attributs de codage de contenu graphique en points.

NOTE – De plus, les trajets et les bords de mise en page doivent être spécifiés afin d'obtenir une rétrocompatibilité avec les profils FOD 026 | PM-26.

De plus amples renseignements concernant les jeux de caractères de profil de document, de commentaires et de présentation de repli sont donnés ci-après.

6.7.4.1 Jeux de caractères de profil

Certains attributs du profil du document ont des valeurs consistant en des chaînes de caractères, par exemple, les attributs de gestion du document. Les jeux de caractères utilisés dans ces chaînes de caractères sont spécifiés par l'attribut de profil de document «jeux de caractères de profil».

L'annonceur d'extension de code et les jeux de caractères désignés et spécifiés par l'attribut «jeux de caractères de profil» sont soumis aux restrictions suivantes:

- L'annonceur d'extension de code doit être 04/03 lorsqu'il est spécifié. Cet annonceur précise l'utilisation des jeux G0 et G1 dans un environnement à 8 bits et aussi l'appel des jeux G0 et G1 dans GL et GR respectivement. Ainsi, dans chaque attribut auquel le présent attribut s'applique les fonctions d'appel ne sont pas nécessaires car les jeux G0 et G1 sont implicitement appelés par cet annonceur d'extension de code.
- *Jeu G0* – La VIR de la Norme ISO 646 (révisée 1991), le jeu primaire de la Norme ISO 6937-2 et une version de la Norme ISO 646, peuvent être désignés pour ce jeu; ces jeux de caractères graphiques ne peuvent être appelés qu'en GL.

- *Jeux G1, G2 et G3* – Il n'est pas imposé de restrictions en ce qui concerne les jeux de caractères graphiques qui peuvent être désignés pour ces jeux; ces jeux de caractères graphiques ne peuvent être appelés qu'en GR.
- Le jeu vide doit être désigné en GL et appelé en GR si aucun autre jeu spécifique n'est appelé en GR.

Si l'attribut «jeux de caractères de profil» n'est pas spécifié, le jeu de caractères désigné et appelé est supposé être le sous-répertoire minimum de Rec. de la série T.410 du CCITT | ISO 8613.

6.7.4.2 Jeux de caractères de commentaires

Les jeux de caractères réputés avoir été désignés et appelés facultativement au début des chaînes de caractères spécifiées par les attributs «commentaires lisibles par l'utilisateur» (voir 6.6.7) et «nom visible par l'utilisateur» (voir 6.6.8) sont indiqués à l'aide de l'attribut de profil du document «jeux de caractères de commentaires».

Cet attribut spécifie également les techniques d'extension de code et les jeux de caractères graphiques qui peuvent être utilisés dans les attributs «commentaires lisibles par l'utilisateur» et «nom visible par l'utilisateur».

Si cet attribut est spécifié, les techniques d'extension de code qui peuvent être utilisées dans les attributs «commentaires lisibles par l'utilisateur» et «nom visible par l'utilisateur» doivent être indiquées par des annonceurs d'extension de code appropriés. L'utilisation du jeu G0 et de GL doit toujours être annoncée. D'autres annonceurs d'extension de code doivent être spécifiés conformément aux exigences d'un document particulier.

Deux techniques d'extension de code sont autorisées pour cet attribut. L'une est d'utiliser GL et GR sans fonctions de commutation de caractères et l'autre est d'utiliser divers jeux de caractères à l'aide de ces fonctions. La première est soumise à des restrictions, mais aucune fonction de commutation de caractères n'est nécessaire dans les attributs «commentaires lisibles par l'utilisateur» et «nom visible par l'utilisateur». Les mêmes restrictions qu'au 6.7.4.1 s'appliquent dans ce cas.

La seconde autorise diverses utilisations de jeux de caractères mais les demandes doivent être spécifiées par des fonctions de commutation de caractères dans les attributs «Commentaires lisibles par l'utilisateur» et «nom visible par l'utilisateur». Les mêmes restrictions qu'au 6.5.1.4 s'appliquent dans ce cas.

Tous les jeux de caractères graphiques qui peuvent être utilisés dans les attributs «commentaires lisibles par l'utilisateur» et «nom visible par l'utilisateur» doivent être désignés dans les «jeux de caractères de commentaires».

Il n'y a pas de restrictions en ce qui concerne les numéros des jeux de caractères graphiques qui sont désignés et/ou appelés dans les «jeux de caractères de commentaires»; par conséquent, la désignation conformément au même jeu G prévaut sur le jeu G précédent.

Si l'attribut «jeux de caractères de commentaires» n'est pas spécifié, le jeu de caractères désigné et appelé est présumé être le sous-répertoire minimum de Rec. de la série T.410 du CCITT | ISO 8613.

6.7.4.3 Jeux de caractères de représentation de remplacement

Cet attribut spécifie les jeux de caractères graphiques désignés et appelés au début de l'attribut «représentation de remplacement» autres que les jeux de caractères graphiques par défaut normalisés.

La restriction imposée aux jeux de caractères du profil décrite au 6.7.4.1 est également appliquée ici. Si cet attribut n'est pas expressément spécifié dans le profil du document, le sous-répertoire minimum de Rec. de la série T.410 du CCITT | ISO 8613 est utilisé dans l'attribut «représentation de remplacement».

6.7.5 Liste de polices

Cette information spécifie toutes les polices (s'il en existe) utilisées dans le document. Elle est signalée à l'aide de l'attribut «liste de polices» (voir B.2).

6.7.6 Attributs de gestion du document

Les attributs de gestion du document renseignent sur le contenu et l'objet du document. Des renseignements sur les points suivants peuvent y être spécifiés:

- description du document (voir la Note);
- dates et heures;
- sources;
- autres informations pour l'utilisateur;
- références externes;

- référence au fichier local;
- attributs du contenu;
- informations de sécurité.

Les attributs qui peuvent être utilisés pour spécifier cette information sont définis dans la Rec. T.414 | ISO 8613-4.

La chaîne de caractères utilisée dans les attributs de gestion du document doit faire partie du jeu de caractères indiqué dans l'attribut de profil de document «jeux de caractères de profil» (voir 6.7.4.1). Si ce dernier attribut n'est pas expressément spécifié dans le profil du document, le jeu de caractères par défaut est alors le sous-répertoire minimum de la Norme ISO 6937-2.

Les fonctions de commande SP, CR et LF peuvent aussi être utilisées en tout point des chaînes de caractères, mais il n'est pas admis d'autres fonctions de commande; le jeu de caractères graphiques ne peut donc pas être modifié dans les attributs de gestion du document.

NOTE – La description du document inclut la référence du document (voir 6.7.2.6).

7 Spécification des contraintes de constitution

Le présent article définit les contraintes de constitution qui peuvent être représentées par des trains de données conformes au présent profil.

7.1 Introduction

Les organigrammes qui montrent les relations entre les constituants des structures logiques sont présentés au 7.1.1. Les macros indiquées sur ces organigrammes sont définies au 7.3.1. Ces macros définissent les valeurs admissibles de l'attribut «générateur de subordonnés» applicables aux constituants et définissent en fait les structures autorisées qui sont admises par le présent profil.

Les organigrammes qui montrent les structures physiques sont présentés au 7.1.2. Les macros indiquées sur ces organigrammes sont définies au 7.4.1.

7.1.1 Diagrammes des relations des constituants logiques

Voir les Figures 21 à 33.

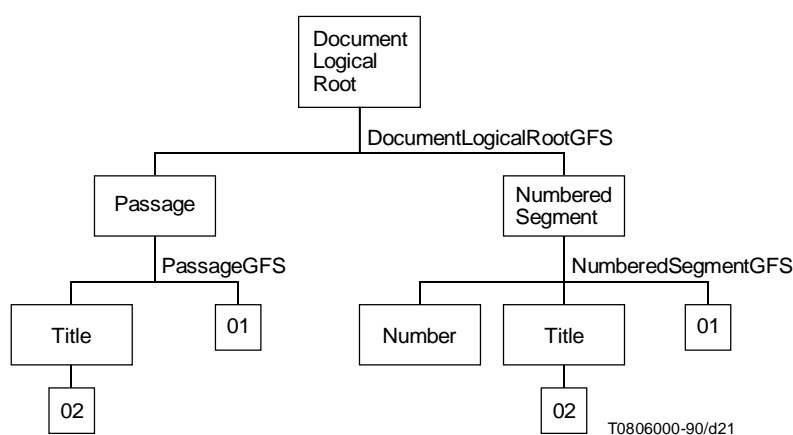
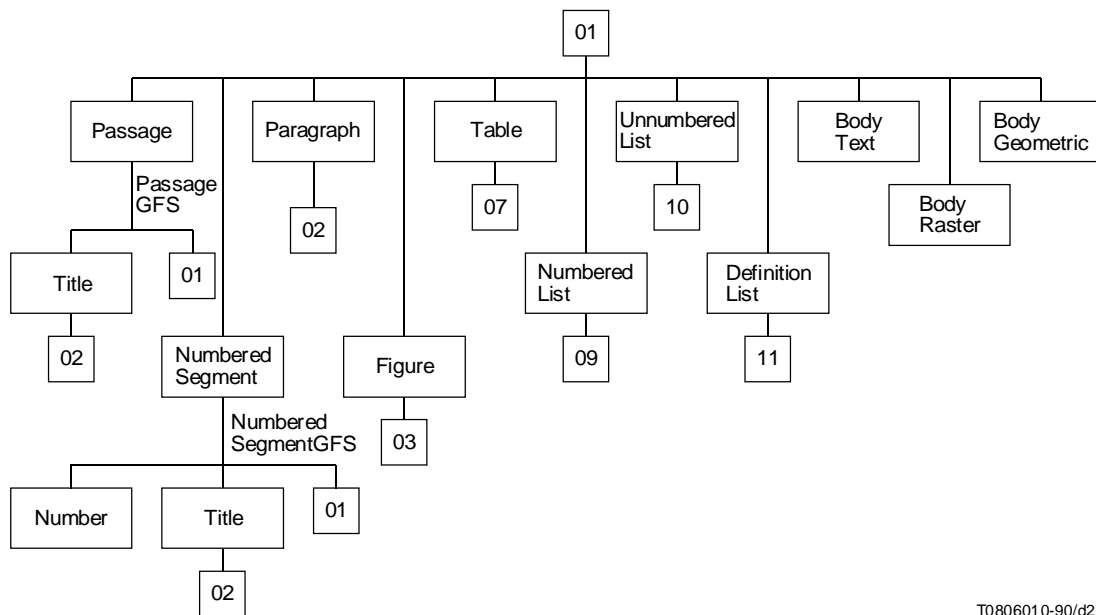
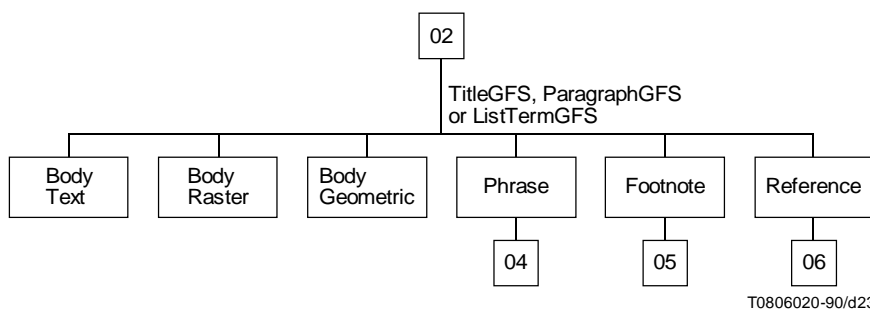


FIGURE 21/T.506
Racine logique de document, 1^{er} niveau



T080601 0-90/d22

FIGURE 22/T.506
Racine logique de document, 2^e niveau



T0806020-90/d23

FIGURE 23/T.506
Titre, paragraphe et terme de liste

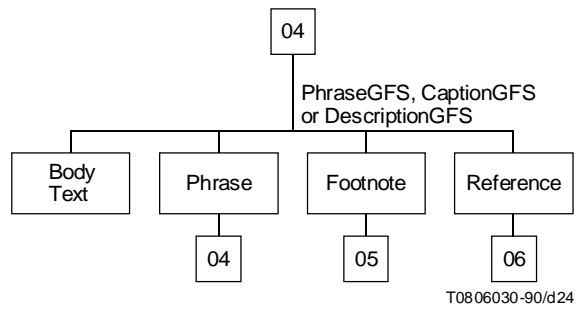


FIGURE 24/T.506
Phrase, légende et description

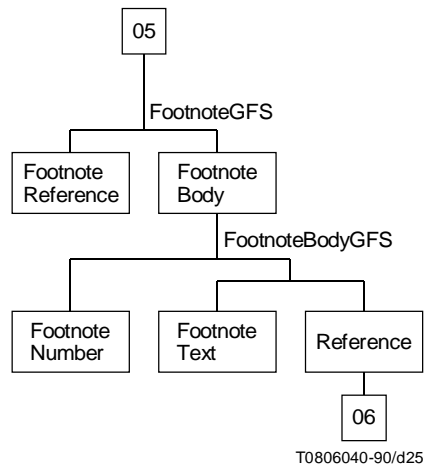


FIGURE 25/T.506
Note de bas de page

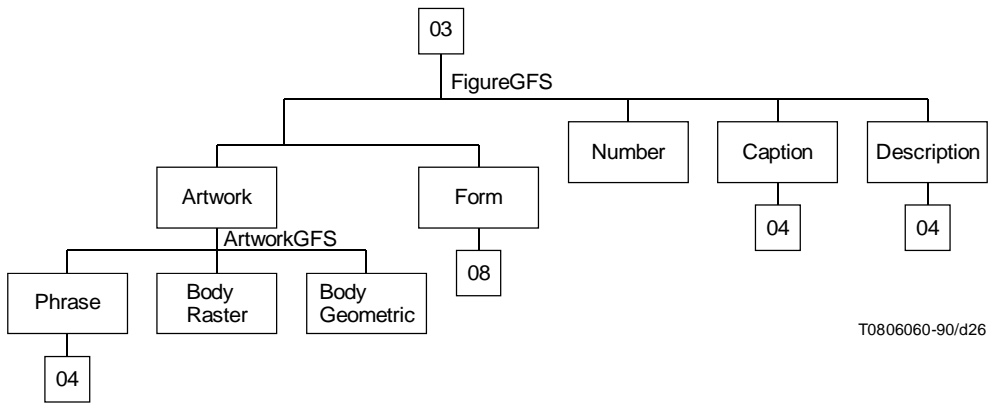


FIGURE 26/T.506
Figure

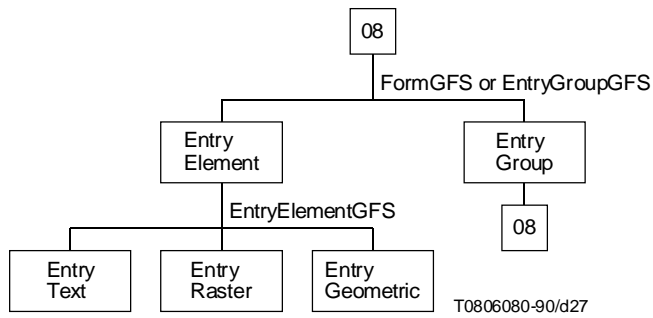


FIGURE 27/T.506
Formulaire

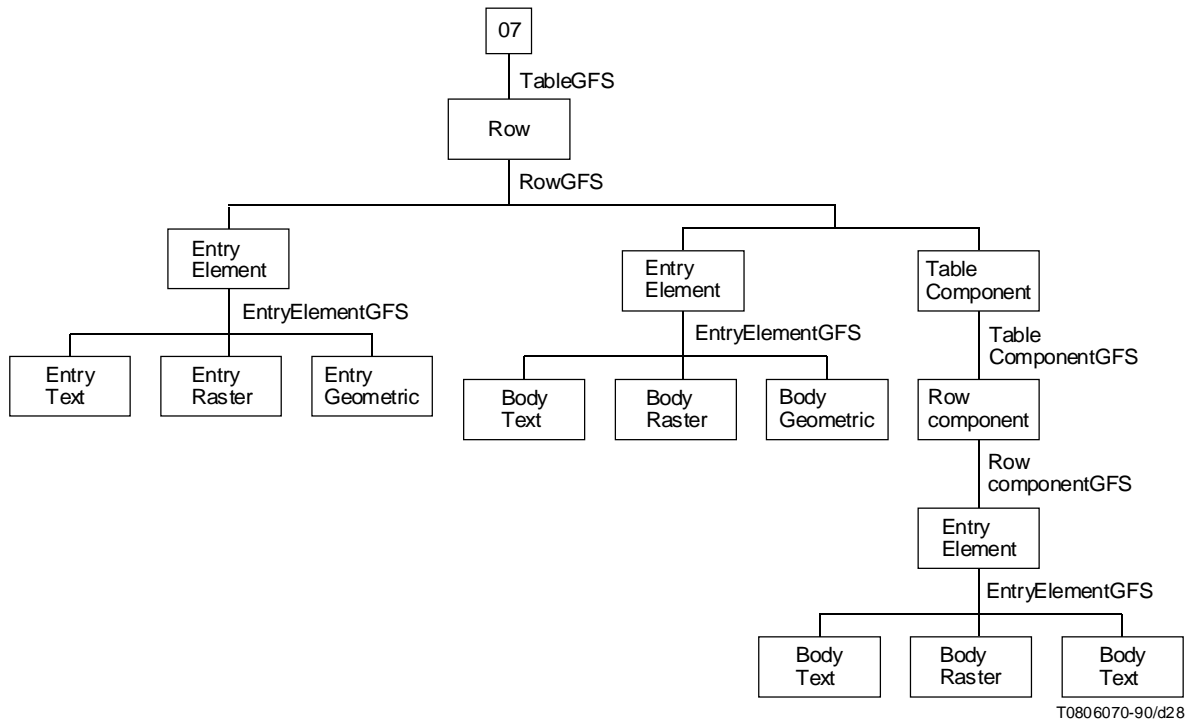


FIGURE 28/T.506
Tableau

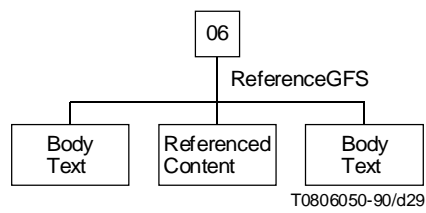


FIGURE 29/T.506
Référence

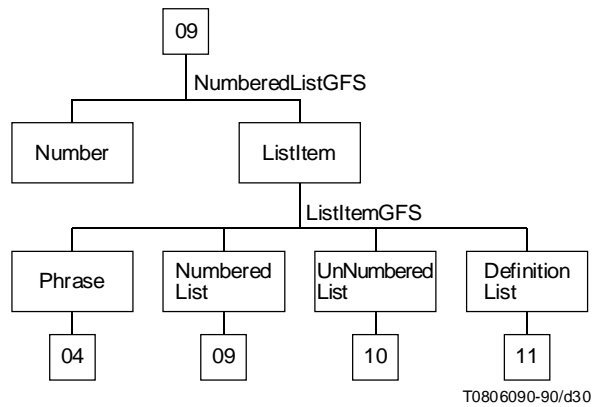


FIGURE 30/T.506
Liste numérotée

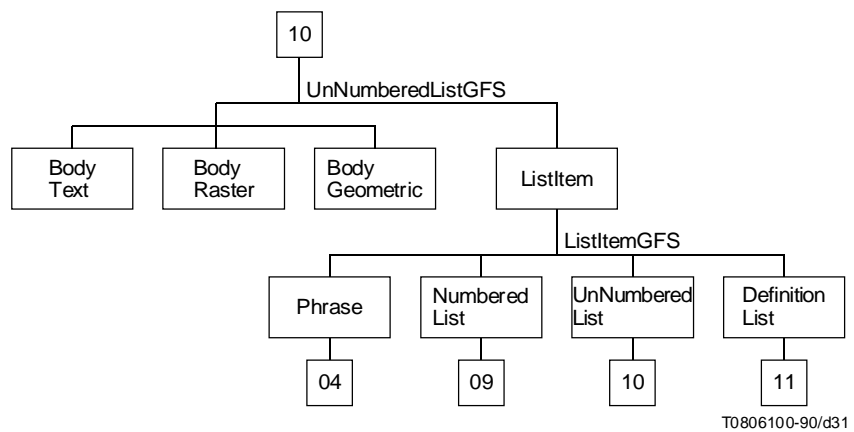


FIGURE 31/T.506
Liste non numérotée

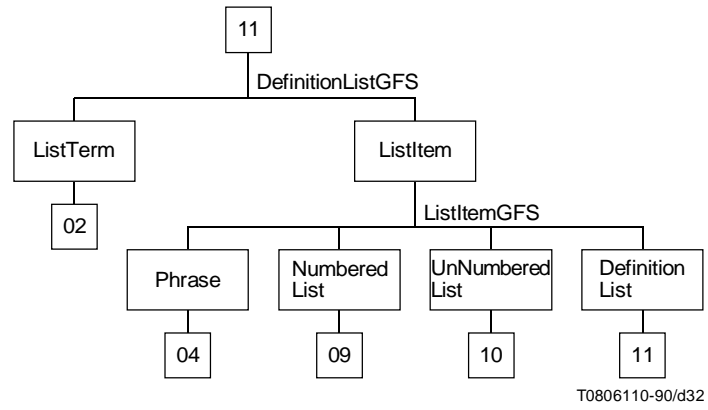


FIGURE 32/T.506
Liste de définitions

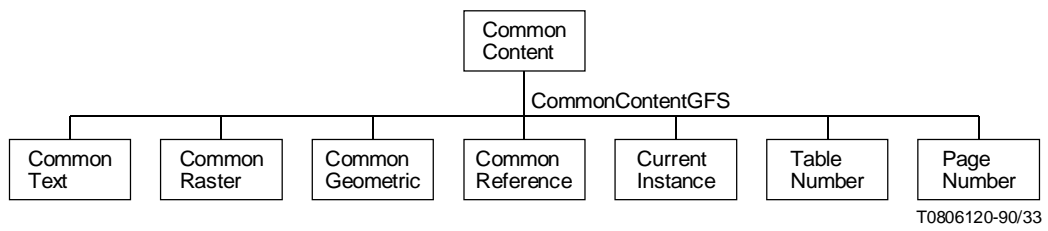


FIGURE 33/T.506
Contenu commun

7.1.2 Diagrammes des relations des constituants physiques

Voir les Figures 34 à 45.

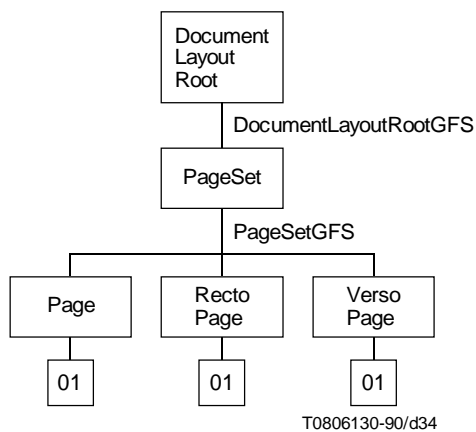


FIGURE 34/T.506
Racine de mise en page de document

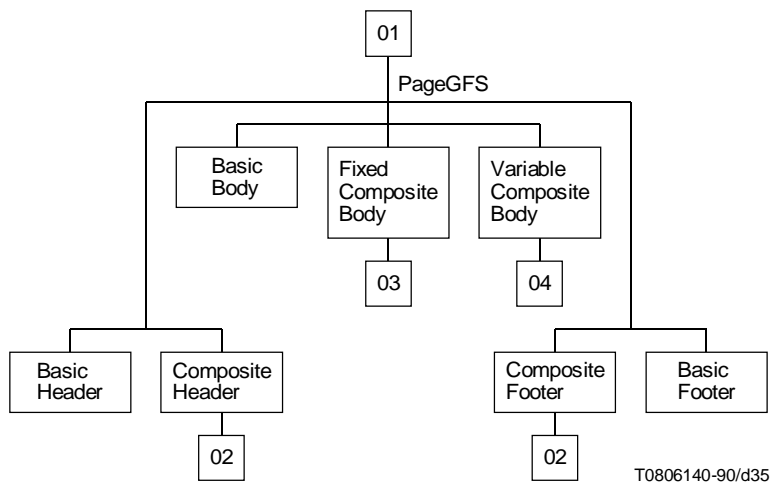


FIGURE 35/T.506
Page recto et page verso

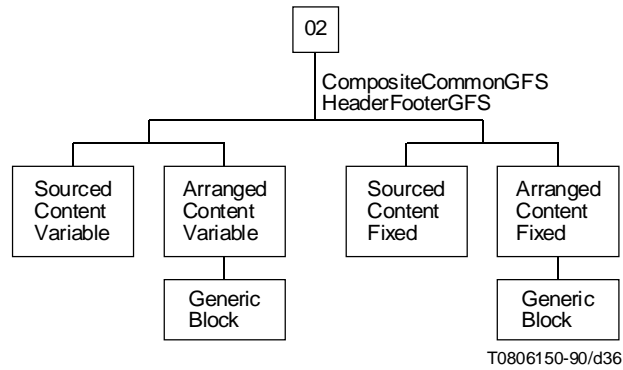


FIGURE 36/T.506

En tête composé, bas de page composé et contenu commun composé

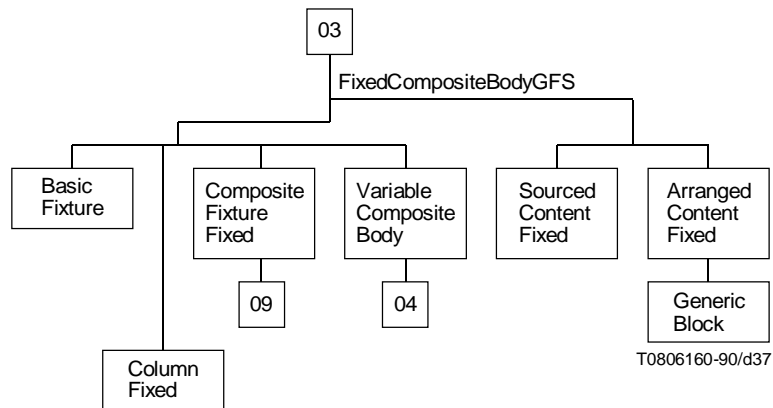


FIGURE 37/T.506

Corps composé fixe

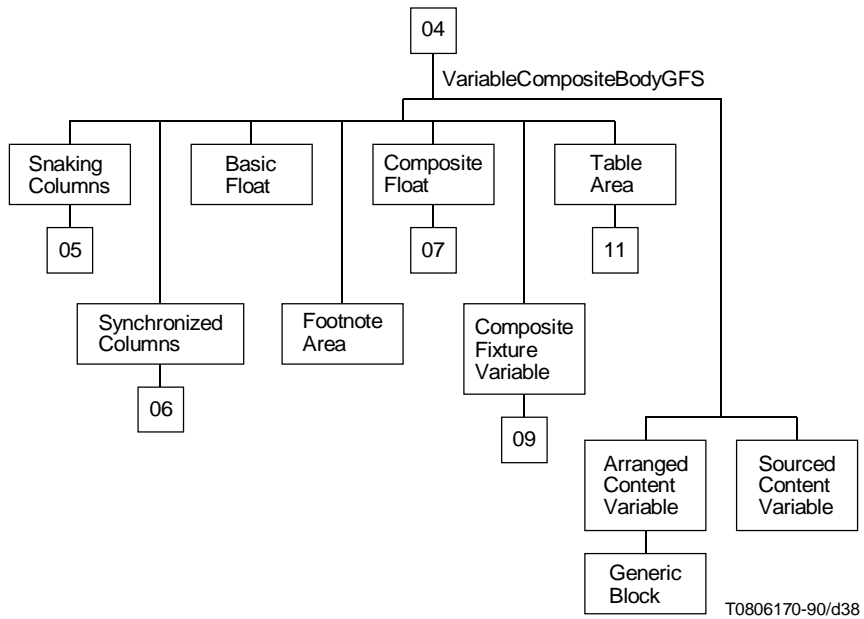


FIGURE 38/T.506
Corps composé variable

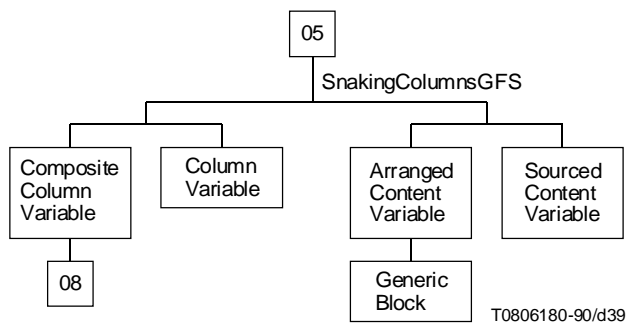


FIGURE 39/T.506
Colonnes serpentes

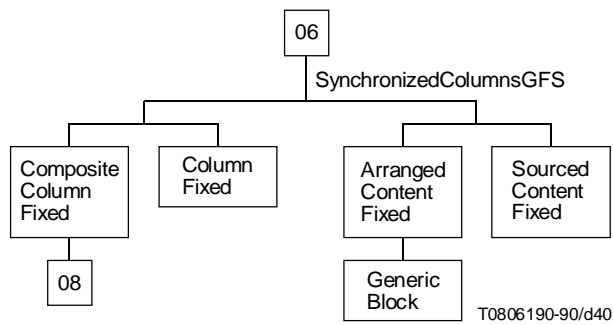


FIGURE 40/T.506
Colonnes synchronisées

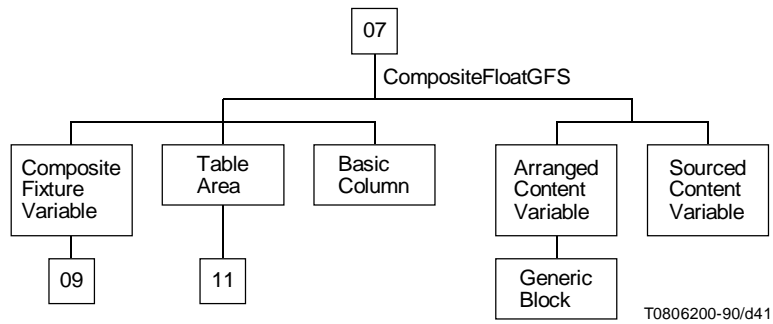


FIGURE 41/T.506
Contenu flottant composé

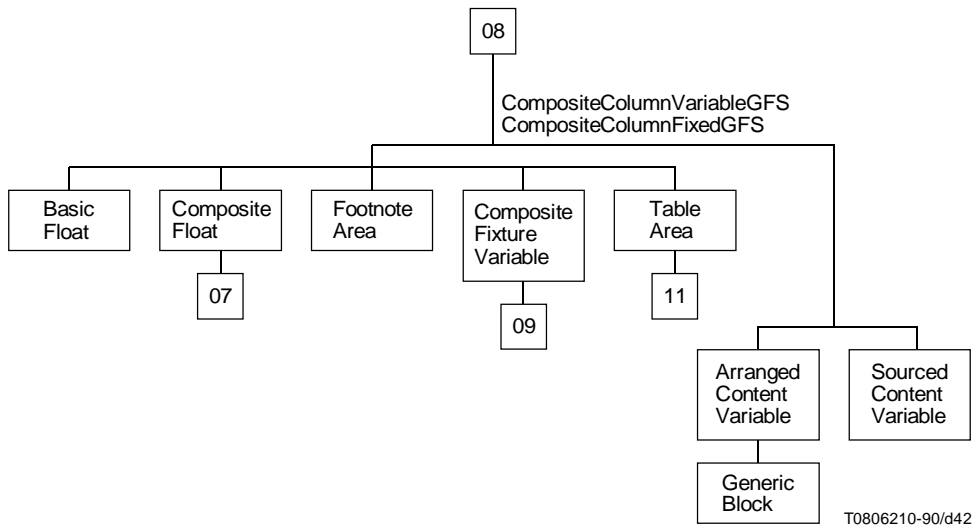


FIGURE 42/T.506
Colonne variable composée et colonne fixe composée

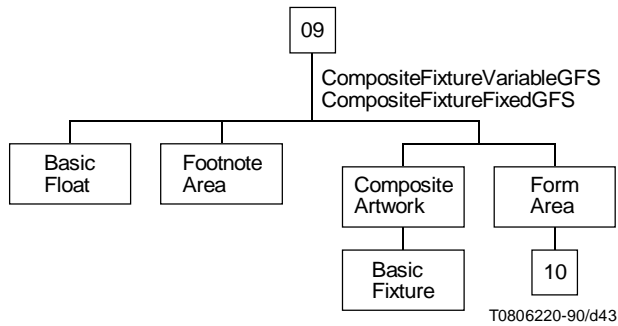


FIGURE 43/T.506
Positionnement composé variable et positionnement composé fixe

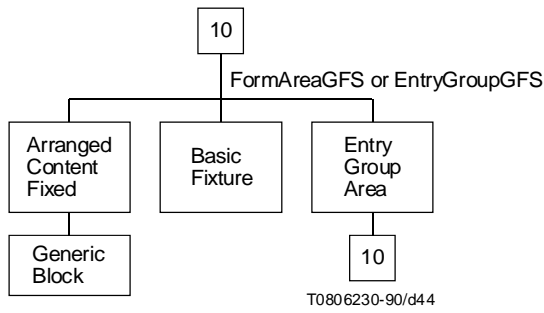


FIGURE 44/T.506
Zone de formulaire et groupe d'entrées

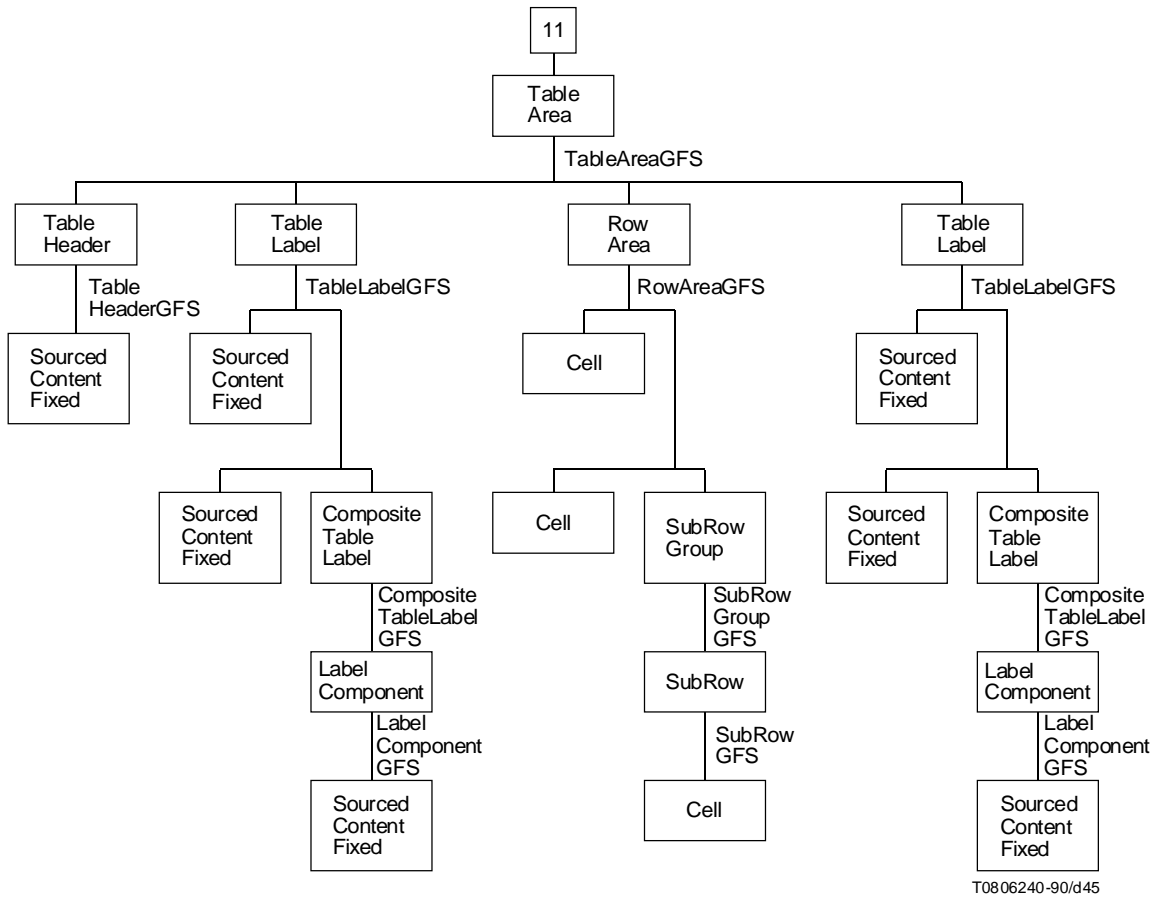


FIGURE 45/T.506
Zone de tableau

7.1.3 Notation

Le présent paragraphe est rédigé conformément à l'Annexe F de la Rec. T.411 du CCITT | ISO 8613-1 «Pro forma et notation correspondante du profil d'application de document» (DAPPN). Certaines extensions mineures et précisions ont néanmoins été apportées, à savoir:

a) [précision]

La définition de la gamme de valeurs pour les attributs «subordonnés» et «ordre de restitution» spécifie l'ensemble des instances d'objet susceptibles d'apparaître. L'ordre et le nombre (qui peut être nul) des instances d'objet pour l'attribut «subordonnés» doivent être conformes avec la valeur de l'attribut «générateur pour subordonnés» dans la classe d'objet correspondante.

b) [précision]

La valeur de «ANY_STRING» peut inclure des fonctions de commande d'extension de code et de caractères graphiques.

c) [extension]

Pour spécifier précisément l'utilisation des jeux de caractères et des fonctions de commande d'extension de code, on utilise les extensions suivantes:

1) Le Tableau 7 définit les symboles utilisés pour désigner les fonctions inversion.

2) L'élément <escape-sequence> est étendu afin d'inclure les fonctions inversion:

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

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

3) La spécification type de données pour #ESC dans l'information de contenu est étendue comme suit:

<escape-sequence>...

d) [précision]

Lorsqu'une valeur d'attribut est spécifiée au moyen d'un ensemble de règles de production, le premier symbole non terminal est son symbole de début. Il convient de noter que l'on utilise des symboles de début autres que <object-id-expr>, <string-expr> et <construction-expr>.

e) [extension]

Les spécifications de type de données autres que celles spécifiées dans les tableaux pour le DAPPN, sont appliquées pour certains attributs dans la plage permise par les normes de base.

f) [extension]

Le caractère «|» est utilisé dans les expressions CASE SUPERIOR pour le format suivant afin d'abrégier le texte:

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

dans laquelle «const1, const2, ..., constn» sont des noms de contraintes sur les constituants et «aaaa» est le nom de l'attribut.

Cette expression est équivalente à l'expression suivante:

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

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

....

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

Lorsque CASE SUPERIOR est évalué, les constituants sont extraits du supérieur immédiat par rapport à la racine. Seul le premier qui satisfait à une des contraintes sur les constituants const1, const2, ..., constn est choisi et l'attribut «aaaa» qu'il contient est testé.

TABLEAU 7/T.506

Symboles désignant les fonctions inversion

Symbole	Fonction inversion	Représentation codée
LS0	Inversion avec blocage 0	00/15
LS1R	Inversion avec blocage un à droite	ESC 07/14
LS2R	Inversion avec blocage deux à droite	ESC 07/13
LS3R	Inversion avec blocage trois à droite	ESC 07/12
SS2	Inversion simple deux	08/14
SS3	Inversion simple trois	08/15

7.2 Document profile constituent constraints

7.2.1 Macro definitions

```

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

```

```

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

```

```

DEFINE(DAC, "DocumentProfile (Document-architecture-class)")
DEFINE(GLAS, "DocumentProfile (Generic-layout-structure)")
DEFINE(COMPLETE, "'complete-generator-set'")

```

```

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

```

-- Any size equal to or smaller than CARA (Common Assured Reproduction Area) of ISO A4 and ANSI A. Both Portrait and Landscape may be specified. Note that the above macro is defined for clarification of the specification and is not used in any other part of this DAP. --

```

DEFINE(NonBasicPageDimensions, "
    REQ #horizontal-dimension
        {REQ #fixed-dimension{<=39680}},
    REQ #vertical-dimension
        {REQ #fixed-dimension {12401..56120}}
|REQ #horizontal-dimension
        {REQ #fixed-dimension{9241..39680}},
    REQ #vertical-dimension
        {REQ #fixed-dimension {<=56120}}
        -- up to ISO A0 portrait --
|REQ #horizontal-dimension
        {REQ #fixed-dimension {<=56120}},
    REQ #vertical-dimension
        {REQ #fixed-dimension {9241..39680}}

```

```

|REQ #horizontal-dimension
  {REQ #fixed-dimension {12401..56120}},
REQ #vertical-dimension
  {REQ #fixed-dimension {<=39680}}
  -- up to ISO A0 landscape --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=40800}},
REQ #vertical-dimension
  {REQ #fixed-dimension {12401..52800}}
|REQ #horizontal-dimension
  {REQ #fixed-dimension {9241..40800}},
REQ #vertical-dimension
  {REQ #fixed-dimension {<=52800}}
  -- up to ANSI E portrait --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=52800}},
REQ #vertical-dimension
  {REQ #fixed-dimension {9241..40800}}
|REQ #horizontal-dimension
  {REQ #fixed-dimension {12401..52800}},
REQ #vertical-dimension
  {REQ #fixed-dimension {<=40800}}
  -- up to ANSI E landscape --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=12141}},
REQ #vertical-dimension
  {REQ #fixed-dimension {12401..17196}}
|REQ #horizontal-dimension
  {REQ #fixed-dimension {9241..12141}},
REQ #vertical-dimension
  {REQ #fixed-dimension {<=17196}}
  -- up to JIS B4(Japanese legal) portrait --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=17196}},
REQ #vertical-dimension
  {REQ #fixed-dimension {9241..12141}}
|REQ #horizontal-dimension
  {REQ #fixed-dimension {12401..17196}},
REQ #vertical-dimension
  {REQ #fixed-dimension {<=12141}}
  -- up to JIS B4(Japanese legal) landscape --

```

)

```

DEFINE(PermissiblePageDimensions, "
  REQ #horizontal-dimension
    {REQ #fixed-dimension {<=39680}},
  REQ #vertical-dimension
    {REQ #fixed-dimension {<=56120}} -- up to ISO A0 portrait --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=56120}},
  REQ #vertical-dimension
    {REQ #fixed-dimension {<=39680}} -- up to ISO A0 landscape --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=40800}},
  REQ #vertical-dimension
    {REQ #fixed-dimension {<=52800}} -- up to ANSI E portrait --
|REQ #horizontal-dimension
  {REQ #fixed-dimension {<=52800}},
  REQ #vertical-dimension
    {REQ #fixed-dimension {<=40800}} -- up to ANSI E landscape --

```

)

```

DEFINE(NominalPageSizes, "
  REQ #horizontal-dimension {7015},      REQ #vertical-dimension {9920}
  -- ISO A5 portrait --
|REQ #horizontal-dimension {9920},      REQ #vertical-dimension {7015}
  -- ISO A5 landscape --

```

REQ #horizontal-dimension {9920},	REQ #vertical-dimension {14030}
REQ #horizontal-dimension {14030},	-- ISO A4 portrait --
REQ #horizontal-dimension {14030},	REQ #vertical-dimension {9920}
REQ #horizontal-dimension {19840},	-- ISO A4 landscape --
REQ #horizontal-dimension {19840},	REQ #vertical-dimension {19840}
REQ #horizontal-dimension {19840},	-- ISO A3 portrait --
REQ #horizontal-dimension {28060},	REQ #vertical-dimension {14030}
REQ #horizontal-dimension {28060},	-- ISO A3 landscape --
REQ #horizontal-dimension {28060},	REQ #vertical-dimension {28060}
REQ #horizontal-dimension {39680},	-- ISO A2 portrait --
REQ #horizontal-dimension {39680},	REQ #vertical-dimension {19840}
REQ #horizontal-dimension {56120},	-- ISO A2 landscape --
REQ #horizontal-dimension {10200},	REQ #vertical-dimension {39680}
REQ #horizontal-dimension {16800},	-- ISO A1 portrait --
REQ #horizontal-dimension {10200},	REQ #vertical-dimension {28060}
REQ #horizontal-dimension {13200},	-- ISO A1 landscape --
REQ #horizontal-dimension {13200},	REQ #vertical-dimension {56120}
REQ #horizontal-dimension {20400},	-- ISO A0 portrait --
REQ #horizontal-dimension {20400},	REQ #vertical-dimension {39680}
REQ #horizontal-dimension {26400},	-- ISO A0 landscape --
REQ #horizontal-dimension {26400},	REQ #vertical-dimension {16800}
REQ #horizontal-dimension {40800},	-- ANSI legal portrait --
REQ #horizontal-dimension {40800},	REQ #vertical-dimension {10200}
REQ #horizontal-dimension {52800},	-- ANSI legal landscape --
REQ #horizontal-dimension {33600},	REQ #vertical-dimension {13200}
REQ #horizontal-dimension {48000},	-- ANSI A portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {10200}
REQ #horizontal-dimension {17196},	-- ANSI A landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {20400}
REQ #horizontal-dimension {12141},	-- ANSI B portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {13200}
REQ #horizontal-dimension {17196},	-- ANSI B landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {26400}
REQ #horizontal-dimension {12141},	-- ANSI C portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {20400}
REQ #horizontal-dimension {17196},	-- ANSI C landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {40800}
REQ #horizontal-dimension {12141},	-- ANSI D portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {26400}
REQ #horizontal-dimension {17196},	-- ANSI D landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {52800}
REQ #horizontal-dimension {12141},	-- ANSI E portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {40800}
REQ #horizontal-dimension {17196},	-- ANSI E landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {48000}
REQ #horizontal-dimension {12141},	-- ANSI F portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {33600}
REQ #horizontal-dimension {17196},	-- ANSI F landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {17196}
REQ #horizontal-dimension {12141},	-- JIS B4 (Japanese legal) portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {12141}
REQ #horizontal-dimension {17196},	-- JIS B4 (Japanese legal) landscape --
REQ #horizontal-dimension {8598},	REQ #vertical-dimension {12141}
REQ #horizontal-dimension {12141},	-- JIS B5 (Japanese letter) portrait --
REQ #horizontal-dimension {12141},	REQ #vertical-dimension {8598}
REQ #horizontal-dimension {17196},	-- JIS B5 (Japanese letter) landscape --

)

```

DEFINE(GRAPHICRENDITIONS, "
  {'cancel'|'increased-intensity'|'decreased-intensity'
  '|italicised'|'underlined'|'slowly-blinking'
  '|rapidly-blinking'|'negative-image'|'crossed-out'
  '|primary-font'|'first-alternative-font'
  '|second-alternative-font'|'third-alternative-font'
  '|fourth-alternative-font'|'fifth-alternative-font'

```

```
'sixth-alternative-font'|'seventh-alternative-font'
'eighth-alternative-font' |'ninth-alternative-font'
'doubly-underlined' | 'normal-intensity' |'not-italicised'
'not-underlined'|'steady'|'positiveimage'
'not-crossed-out' }... ")
```

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

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

-- Macro defining code extension announcers for DAP defaults --

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

-- Macros defining final character for designation --

```
DEFINE(FCORE, "04/02 -- A final character designating ISO IR 6 (the IRV of ISO/IEC 646, i.e. ASCII) --")
DEFINE(F646, " -- A final character designating any version of ISO/IEC 646 except ISO IR 6 --")
DEFINE(F94S, " -- A final character designating any registered 94 single byte graphic character set, optionally
preceded by one or more intermediate characters as defined in Annex C ISO 2022 --")
DEFINE(F94M, " -- A final character designating any registered 94 multi-byte graphic character set, optionally
preceded by one or more intermediate characters as defined in Annex C ISO 2022 --")
DEFINE(F96S, " -- A final character designating any registered 96 single byte graphic character set, optionally
preceded by one or more intermediate characters as defined in Annex C ISO 2022 --")
DEFINE(F96M, " -- A final character designating any registered 96 multi byte graphic character set, optionally
preceded by one or more intermediate characters as defined in Annex C ISO 2022 --")
DEFINE(FEMPTY, "07/14 -- The empty set --")
```

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

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

-- Macros defining designation sequences --

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

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

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

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

```

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

```

-- Designate any character set to G3 --

```

DEFINE(DEG-EMPTY-G1, "ESC 02/09 $FEMPTY")

```

-- Designate the empty set to G1 --

-- Macro defining permissible graphic character sets --

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

-- Macro defining comments character sets --

```

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

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

```

```

{ $DEG-CORE-G0 | $DEG-646-G0 } -- designate G0 --

{{ $DEG-ANY-G1
| $DEG-ANY-G2
| $DEG-ANY-G3 }... -- designate G3 --
| $DEG-EMPTY-G1 }

```

)

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

DEFINE(ALTCHAR, "\$PROFCHAR")

7.2.2 Constituent constraints

7.2.2.1 DocumentProfile

```

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

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

  $FPDA: PERM Generic-layout-structure {'complete-generator-set'},
        -- shall be present if there is no external document class reference --
        PERM Specific-layout-structure {'present'},
        -- shall be present in the case of specific document and shall not be present in the case of generic
        document --
        PERM Generic-logical-structure {'complete-generator-set'
                                        |'partial-generator-set'},
        -- shall be present if there is no external document class reference --
        PERM Specific-logical-structure {'present'},
        -- shall be present in the case of complete document and shall not be present in the case of generic
        document --
        PERM Presentation-styles {'present'},
        PERM Layout-styles {'present'}
},
PERM External-document-class {ANY_VALUE},
PERM Resource-document {ANY_VALUE},
PERM Resources {MUL {REQ #resource-identifier {ANY_VALUE},
                    REQ #resource-object-class-identifier {ANY_VALUE}}
},

-- document characteristics --
REQ Document-application-profile -- See clause 8 for a definition of the permitted values for this
attribute --},

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

```

```

PERM #dimensions      {$PermissiblePageDimensions},
PERM #medium-type    {PERM #nominal-page-size {$NominalPageSizes},
PERM #side-of-sheet   {ANY_VALUE }},
PERM #transparency   {ANY_VALUE},
PERM #colour         {ANY_VALUE},
PERM #layout-path    {ANY_VALUE},
PERM #block-alignment {ANY_VALUE},
PERM #border         {ANY_VALUE},
PERM #page-position  {ANY_VALUE},
PERM #type-of-coding
    {ASN.1 {2 8 3 6 0} -- character encoding --
    |ASN.1 {2 8 3 7 0} -- T.6 encoding --
    |ASN.1 {2 8 3 7 1} -- T.4 one dimensional encoding --
    |ASN.1 {2 8 3 7 2} -- T.4 two dimensional encoding --
    |ASN.1 {2 8 3 7 3} -- bitmap encoding --
    |ASN.1 {2 8 3 8 0} -- geometric encoding --},
PERM #character-content-defaults {
    PERM #alignment           {ANY_VALUE},
    PERM #character-fonts     {ANY_VALUE},
    PERM #character-path      {ANY_VALUE},
    PERM #character-spacing    {ANY_VALUE},
    PERM #character-orientation {ANY_VALUE},
    PERM #code-extension-announcers {$DAP-DEFAULT-CDEXTEN},
    PERM #first-line-offset    {ANY_VALUE},
    PERM #graphic-character-sets {$DAP-DEFAULT-GRCHAR},
    PERM #graphic-character-subrepertoire {ANY_VALUE},
    PERM #graphic-rendition    {$GRAPHICRENDITIONS},
    PERM #itemization         {ANY_VALUE},
    PERM #kerning-offset      {ANY_VALUE},
    PERM #line-layout-table    {ANY_VALUE},
    PERM #line-progression    {ANY_VALUE},
    PERM #line-spacing       {ANY_VALUE},
    PERM #pairwise-kerning    {ANY_VALUE},
    PERM #indentation        {ANY_VALUE},
    PERM #orphan-size        {ANY_VALUE},
    PERM #proportional-line-spacing {ANY_VALUE},
    PERM #widow-size         {ANY_VALUE},
    PERM #formatting-indicator {ANY_VALUE},
    PERM #initial-offset     {ANY_VALUE}
},
PERM #raster-graphics-content-defaults {
    PERM #pel-path           {ANY_VALUE},
    PERM #line-progression    {ANY_VALUE},
    PERM #image-dimensions    {ANY_VALUE},
    PERM #pel-spacing        {ANY_VALUE},
    PERM #spacing-ratio      {ANY_VALUE},
    PERM #compression       {ANY_VALUE}
},
PERM #geometric-graphics-content-defaults {ANY_VALUE}
},
REQ Document-architecture-class      {$FDA| $PDA |$FPDA},
REQ Content-architecture-classes     {[$FC],[ $PC],[ $FPC],[ $FPR],[ $FPG]},
REQ Interchange-format-class         {-- See clause 8 for the definition of the permitted values for this attribute. --},
REQ Oda-version                      {REQ #standard-or-recommendation
    {"CCITT Recs. T.410-Series (1988) | ISO 8613 (1989); version 1.1"},
    REQ #publication-date             {"1992-01-01"}},

```

-- non basic document characteristics --

```

PERM   Profile-character-sets           { $PROFCHAR },
PERM   Comments-character-sets         { $COMCHAR },
PERM   Alternative-representation-character-sets { $ALTCHAR },
PERM   Page-dimensions   { PMUL { $NonBasicPageDimensions },
PERM   Medium-types      { PMUL { PERM #nominal-page-size      { $NominalPageSizes,
                                PERM #side-of-sheet          { 'recto'|'verso' } },
    -- All values of "medium type" are non-basic --
PERM Layout-paths      { { '0-degrees'|'90-degrees'|'180-degrees' } ... },
    -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --
PERM   Borders         { ANY_VALUE },
    -- Any values need not be declared, they may be specified here for upwards compatibility from FOD26 --
PERM   Coding-attributes {
PERM   #raster-graphics-coding-attributes {
    PERM #compression      { 'uncompressed' }
}
}
PERM   Presentation-features {
    PERM #character-presentation-features {
        PERM #character-orientation      { '90-degrees' },
        -- This value need not be declared, they may be specified here for upwards compatibility from FOD26 --
        PMUL { PERM #character-path          { '90-degrees'|'180-degrees'|'270-degrees' },
        -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --
        PMUL { PERM #character-spacing      { <100 | 100 | 160 | 200 },
        -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --
        PMUL { PERM #graphic-rendition      { 'crossed-out'|'not-crossed-out' },
        -- These values need not be declared, they may be specified here for upwards compatibility from FOD26 --
        PMUL { PERM #line-spacing          { ANY_VALUE } EXCEPT { 200, 300, 400 },
        -- Values 100,150 and 600 need not to be declared, they may be specified here for upwards compatibility
        -- from FOD26 --
        PERM #line-progression           { '90-degrees' },
        -- This value need not be declared, they may be specified here for upwards compatibility from FOD26 --
        PMUL { PERM #graphic-character-sets { $NON-BASIC-GRCHAR },
        PMUL { PERM #graphic-character-subrepertoire { ANY_VALUE } }
}
}
PERM #raster-graphics-presentation-features {
    PMUL { PERM #pel-path          { '90-degrees'|'180-degrees'|'270-degrees' },
    PERM #line-progression        { '90-degrees' },
    PMUL { PERM #pel-spacing      { ANY_VALUE } EXCEPT { 16, 12, 8, 6, 5, 4, 3, 2, 1 },
    -- Any value of #pel spaces is permitted as basic --
    -- Basic values of #length are multiples of #pel spaces as listed --
    PMUL { PERM #spacing-ratio
        { REQ #line-spacing-value { ANY_VALUE } EXCEPT { 1,
        REQ #pel-spacing-value   { ANY_VALUE } EXCEPT { 1 } }
}
}
}
    -- additional document characteristics --
PERM Unit-scaling      { REQ #unit-scaling-m { ANY_INTEGER,
                                REQ #unit-scaling-n { ANY_INTEGER },
PERM Fonts-list      { PMUL { REQ #font-identifier { ANY_VALUE,
                                REQ #font-reference { ANY_VALUE } }
}
}

```


-- document management attributes --

-- document description --

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

-- dates and times --

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

-- originators --

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

-- other user information --

PERM Copyright	{ANY_VALUE},
PERM Status	{ANY_STRING},
PERM User-specific-codes	{ANY_STRING...},
PERM Distribution-list	{ANY_VALUE},
PERM Additional-information	{ANY_VALUE},

-- external references --

PERM References-to-other-documents	{ANY_VALUE},
PERM Superseded-documents	{ANY_VALUE},

-- local file references --

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

-- content attributes --

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

-- security information --

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

}

7.3 Logical constituent constraints

7.3.1 Macro definitions

-- Defines any logical objects in a specific logical structure. --

```
DEFINE(LogicalObjects, "  
<logical-objects> ::= OBJECT_ID_OF(DocumentLogicalRoot)  
| OBJECT_ID_OF(Passage)  
| OBJECT_ID_OF(NumberedSegment)  
| OBJECT_ID_OF(Number)  
| OBJECT_ID_OF(Title)  
| OBJECT_ID_OF(Caption)  
| OBJECT_ID_OF(Paragraph)  
| OBJECT_ID_OF(Phrase)  
| OBJECT_ID_OF(Footnote)  
| OBJECT_ID_OF(FootnoteNumber)
```

```

| OBJECT_ID_OF(FootnoteReference)
| OBJECT_ID_OF(FootnoteBody)
| OBJECT_ID_OF(FootnoteText)
| OBJECT_ID_OF(Figure)
| OBJECT_ID_OF(BodyText)
| OBJECT_ID_OF(Reference)
| OBJECT_ID_OF(ReferencedContent)
| OBJECT_ID_OF(BodyRaster)
| OBJECT_ID_OF(BodyGeometric)
| OBJECT_ID_OF(Description)
| OBJECT_ID_OF(Artwork)
| OBJECT_ID_OF(NumberedList)
| OBJECT_ID_OF(UnNumberedList)
| OBJECT_ID_OF(DefinitionList)
| OBJECT_ID_OF(ListItem)
| OBJECT_ID_OF(ListTerm)
| OBJECT_ID_OF(Table)
| OBJECT_ID_OF(Row)
| OBJECT_ID_OF(TableComponent)
| OBJECT_ID_OF(RowComponent)
| OBJECT_ID_OF(Form)
| OBJECT_ID_OF(EntryElement)
| OBJECT_ID_OF(EntryGroup)
| OBJECT_ID_OF(EntryText)
| OBJECT_ID_OF(EntryRaster)
| OBJECT_ID_OF(EntryGeometric);

```

)

-- Defines any logical object classes other than classes referred by logical source. --

DEFINE(LogicalObjectClasses, "

```

<logical-object-classes> ::= OBJECT_CLASS_ID_OF(DocumentLogicalRoot)
| OBJECT_CLASS_ID_OF(Passage)
| OBJECT_CLASS_ID_OF(NumberedSegment)
| OBJECT_CLASS_ID_OF(Number)
| OBJECT_CLASS_ID_OF(Title)
| OBJECT_CLASS_ID_OF(Caption)
| OBJECT_CLASS_ID_OF(Paragraph)
| OBJECT_CLASS_ID_OF(Phrase)
| OBJECT_CLASS_ID_OF(Footnote)
| OBJECT_CLASS_ID_OF(FootnoteNumber)
| OBJECT_CLASS_ID_OF(FootnoteReference)
| OBJECT_CLASS_ID_OF(FootnoteBody)
| OBJECT_CLASS_ID_OF(FootnoteText)
| OBJECT_CLASS_ID_OF(Figure)
| OBJECT_CLASS_ID_OF(BodyText)
| OBJECT_CLASS_ID_OF(Reference)
| OBJECT_CLASS_ID_OF(ReferencedContent)
| OBJECT_CLASS_ID_OF(BodyRaster)
| OBJECT_CLASS_ID_OF(BodyGeometric)
| OBJECT_CLASS_ID_OF(Description)
| OBJECT_CLASS_ID_OF(Artwork)
| OBJECT_CLASS_ID_OF(NumberedList)
| OBJECT_CLASS_ID_OF(UnNumberedList)
| OBJECT_CLASS_ID_OF(DefinitionList)
| OBJECT_CLASS_ID_OF(ListItem)
| OBJECT_CLASS_ID_OF(ListTerm)
| OBJECT_CLASS_ID_OF(Table)
| OBJECT_CLASS_ID_OF(Row)
| OBJECT_CLASS_ID_OF(TableComponent)
| OBJECT_CLASS_ID_OF(RowComponent)
| OBJECT_CLASS_ID_OF(Form)
| OBJECT_CLASS_ID_OF(EntryElement)
| OBJECT_CLASS_ID_OF(EntryGroup)
| OBJECT_CLASS_ID_OF(EntryText)
| OBJECT_CLASS_ID_OF(EntryRaster)
| OBJECT_CLASS_ID_OF(EntryGeometric);

```

)

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

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

```
DEFINE(PREFIX, "
<prefix> ::= ''prefix-''<n>;
$N
")
```

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

```
DEFINE(SUFFIX, "
<suffix> ::= ''suffix-''<n>;
$N
")
```

-- Defines the separator binding. This binding is used to provide a separator character for a hierarchical form of a segment number, figure number, table number, footnote number or page number. The instances are differentiated by a suffix number. --

```
DEFINE(SEPARATOR, "
<separator> ::= ''separator-''<n>;
$N
")
```

-- Defines the general number binding. This binding may be instanced for use as the numeric value such as in segment number, figure number, table number, list number, footnote number or page number bindings. The instances are differentiated by a suffix number. --

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

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

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

-- Defines the general string binding. The instances are differentiated by a suffix number. --

```
DEFINE(STRING, "
<string> ::= ''string-''<n>;
$N
")
```

-- Defines the names for footnote categories. The instances are differentiated by a suffix number. --

```
DEFINE(FOOTNOTECATEGORY, "
<footnotecategory> ::= ''Footnote''[''-''<n>];
$N
")
```

```
DEFINE(INITIALISEBINDINGS, "
REQ #binding-name      {$PREFIX},
REQ #binding-value     {ANY_STRING}
|REQ #binding-name     {$SUFFIX},
REQ #binding-value     {ANY_STRING}
|REQ #binding-name     {$SEPARATOR},
```

```

    REQ #binding-value          {ANY_STRING}
|REQ #binding-name            {$NUMBER},
    REQ #binding-value          {ANY_INTEGER}
|REQ #binding-name            {$NUMBERSTRING},
    REQ #binding-value          {ANY_STRING}
|REQ #binding-name            {$STRING},
    REQ #binding-value          {ANY_STRING}
")
-- Used to make a simple or compound string out of the number bindings. --

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

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

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

$NUMBERSTRING
$SEPARATOR
$NUMBER}
")
-- Used to increment any of the number bindings. --

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

$NUMBER}
")
-- Used to initialize/specify or manipulate the bindings. The bindings defined by this macro are permitted to:
- any logical object class,
- any logical object,
- any layout object class except frame classes and block classes.
--

DEFINE(SPECIFYBINDINGS, "
$INITIALISEBINDINGS | $USENUMBERS | $USENUMBERSTRINGS
")
-- Used to initialize fnotenum and fnotestring bindings. --
-- Note that footnote numbering is realized as a particular case of the general number binding mechanism supported by
this DAP using the bindings <number> and <numberstring>. "fnotenum" and "fnotestring" may be used for the
compatibility with FOD26. --

DEFINE(INITIALISEFNOTE, "
    REQ #binding-name          {""fnotenum""},
    REQ #binding-value          {>=0}
")
-- Used to increment fnotenum binding. --

```

```

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

```

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

```

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

```

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

```

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

```

-- Used to initialise PGnum binding. --

-- Note that a page numbering is realized as a particular case of the general number binding mechanism supported by this DAP using the bindings <number> and <numberstring>. "PGnum" may be used for the compatibility with FOD26. --

```

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

```

-- Used to increment PGnum binding. --

```

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

```

-- This string expression is allowed in a content generator for Number to automatically generate text for segment numbers, figure numbers or list numbers. (Note: B_REF(CURR-OBJ) is used for list numbers.) --

```

DEFINE(SEGMENTNUMBER, "
<string-expr> ::= [<pre-str><num-str>[<suf-str>];
<num-str>      ::= B_REF(SUP(CURR-OBJ))(<numberstring>)
                | B_REF(CURR-OBJ)(<numberstring>);
<pre-str>     ::= B_REF(SUP(CURR-OBJ))(<prefix>) | ANY_STRING;
<suf-str>     ::= B_REF(SUP(CURR-OBJ))(<suffix>) | ANY_STRING;
$NUMBERSTRING
$PREFIX
$SUFFIX
")

```

-- This string expression is allowed in a content generator for TableNumber to automatically generate text for a table number. --

```

DEFINE(TABLENUMBER, "
<string-expr> ::= [<pre-str><num-str>[<suf-str>];
<num-str>     ::= B_REF
                (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ)))
                (<numberstring>);
<pre-str>    ::= B_REF(SUP
                (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ)))
                (<prefix>)
                | ANY_STRING;
")

```

```

<suf-str> ::= B_REF(SUP
              (CURR-INST(OBJECT_CLASS_ID_OF(Table),(CURR-OBJ)))
              (<suffix>)
              | ANY_STRING;
$NUMBERSTRING
$PREFIX
$SUFFIX
")

```

-- This string expression is allowed in a content generator for PageNumber to automatically generate text for a page number. --

-- Note that a page number may be generated either from <number> or <numberstring> or from PGnum. PGnum is kept for the compatibility with FOD26. --

```

DEFINE(PGNUMBER, "
<string-expr> ::= [<pre-str>]<num-str>[<suf-str>];
<pre-str> ::= B_REF(SUP(<current-layout-object>))(<prefix>) | ANY_STRING;
<suf-str> ::= B_REF(SUP(<current-layout-object>))(<suffix>) | ANY_STRING;

<num-str> ::= MK-STR(<numeric-expr>)
           | U-ALPHA(<numeric-expr>)
           | L-ALPHA(<numeric-expr>)
           | U-ROM(<numeric-expr>)
           | L-ROM(<numeric-expr>)
           | B_REF(SUP(<layout-object-1>))(<numberstring>)
           | B_REF(<layout-object-2>)(<numberstring>);

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

<current-layout-object> ::= <layout-object-1> | <layout-object-2>;
<layout-object-1> ::= CURR-INST(<class-or-type-1>,(CURR-OBJ));
<layout-object-2> ::= CURR-INST(<class-or-type-2>,(CURR-OBJ));
<class-or-type-1> ::= 'frame';
<class-or-type-2> ::= 'page'
                 | OBJECT_CLASS_ID_OF(Page)
                 | OBJECT_CLASS_ID_OF(RectoPage)
                 | OBJECT_CLASS_ID_OF(VersoPage);

$PREFIX
$SUFFIX
$NUMBER
$NUMBERSTRING
")

```

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

-- Note that a footnote number may be generated either from <numberstring> or from "fnotestring". "fnotestring" is kept for the compatibility with FOD26. --

```

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

$NUMBERSTRING
$PREFIX
$SUFFIX
")

```

-- This string expression is allowed in a content generator for ReferencedContent to automatically generate text for references such as to segment numbers, table numbers, figure numbers, list numbers, footnote numbers and <string> bindings associated with a referring (i.e. a target) logical object, or to page numbers, pageset numbers and <string> bindings associated with a layout object in which the referring logical object is laid out. --

DEFINE(REF, "
<string-expr> ::= [<pre-str>]<ref-str>[<suf-str>];

-- These are a prefix and a suffix of ReferencedContent itself, not those of referring text. e.g. '(See' and ')'. --

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

<ref-str> ::= { <ref-numberstring>
| <ref-fnotestring>
| <ref-pgnum>
| <ref-number>
| <ref-string>
| ANY_STRING }... ;

<ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];

<pre-str-a> ::= B_REF(SUP(<target-object-1>))(<prefix>)
| B_REF(<target-object>)(<prefix>);

<suf-str-a> ::= B_REF(SUP(<target-object-1>))(<suffix>)
| B_REF(<target-object>)(<suffix>);

<ref-str-a> ::= B_REF(SUP(<target-object-1>))(<numberstring>)
| B_REF(<target-object>)(<numberstring>);

<ref-fnotestring> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];

<pre-str-b> ::= B_REF(SUP(<target-logical-object-1>))(<prefix>)
| B_REF(<target-logical-object>)(<prefix>);

<suf-str-b> ::= B_REF(SUP(<target-logical-object-1>))(<suffix>)
| B_REF(<target-logical-object>)(<suffix>);

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

<ref-pgnum> ::= [<pre-str-c>] <ref-str-c> [<suf-str-c>];

<pre-str-c> ::= B_REF(SUP(<target-layout-object>))(<prefix>);

<suf-str-c> ::= B_REF(SUP(<target-layout-object>))(<suffix>);

<ref-str-c> ::= MK-STR(B_REF(SUP(<layout-object-1>))("PGnum"))
| U-ALPHA(B_REF(SUP(<layout-object-1>))("PGnum"))
| L-ALPHA(B_REF(SUP(<layout-object-1>))("PGnum"))
| U-ROM(B_REF(SUP(<layout-object-1>))("PGnum"))
| L-ROM(B_REF(SUP(<layout-object-1>))("PGnum"))
| MK-STR(B_REF(<layout-object-2>))("PGnum"))
| U-ALPHA(B_REF(<layout-object-2>))("PGnum"))
| L-ALPHA(B_REF(<layout-object-2>))("PGnum"))
| U-ROM(B_REF(<layout-object-2>))("PGnum"))
| L-ROM(B_REF(<layout-object-2>))("PGnum"));

<ref-number> ::= [<pre-str-d>] <ref-str-d> [<suf-str-d>];

<pre-str-d> ::= B_REF(SUP(<target-object-1>))(<prefix>)
| B_REF(<target-object>)(<prefix>);

<suf-str-d> ::= B_REF(SUP(<target-object-1>))(<suffix>)
| B_REF(<target-object>)(<suffix>);

<ref-str-d> ::= MK-STR(B_REF(SUP(<target-object-1>))(<number>))
| U-ALPHA(B_REF(SUP(<target-object-1>))(<number>))
| L-ALPHA(B_REF(SUP(<target-object-1>))(<number>))
| U-ROM(B_REF(SUP(<target-object-1>))(<number>))
| L-ROM(B_REF(SUP(<target-object-1>))(<number>))
| MK-STR(B_REF(<target-object>)(<number>))
| U-ALPHA(B_REF(<target-object>)(<number>))
| L-ALPHA(B_REF(<target-object>)(<number>))
| U-ROM(B_REF(<target-object>)(<number>))
| L-ROM(B_REF(<target-object>)(<number>));

<ref-string> ::= [<pre-str-e>] <ref-str-e> [<suf-str-e>];

<pre-str-e> ::= B_REF(SUP(<target-object-1>))(<prefix>)
| B_REF(<target-object>)(<prefix>);

<suf-str-e> ::= B_REF(SUP(<target-object-1>))(<suffix>)
| B_REF(<target-object>)(<suffix>);

<ref-str-e> ::= B_REF(SUP(<target-object-1>))(<string>)
| B_REF(<target-object>)(<string>);

```

<target-object> ::= <target-logical-object> | <target-layout-object>;
<target-object-1> ::= <target-logical-object-1> | <target-layout-object>;

<target-logical-object> ::= <logical-objects> | CURR-INST(<class-or-type-logical>,<logical-objects>);
<target-logical-object-x> ::= <logical-objects> | CURR-INST(<class-or-type-logical>,<logical-objects>);
<target-logical-object-1> ::= CURR-INST(<class-or-type-logical>,<logical-objects>);
<class-or-type-logical> ::= <logical-object-classes>
    | 'composite-logical-object'
    | 'basic-logical-object';

<target-layout-object> ::= <layout-object-1> | <layout-object-2>;
<layout-object-1> ::= CURR-INST(<class-or-type-layout-1>,<target-logical-object-x>);
<layout-object-2> ::= CURR-INST(<class-or-type-layout-2>,<target-logical-object-x>);
<class-or-type-layout-1> : <class-or-type-layout-1> ::= 'frame';
<class-or-type-layout-2> ::= 'page'
    | OBJECT_CLASS_ID_OF(Page)
    | OBJECT_CLASS_ID_OF(RectoPage)
    | OBJECT_CLASS_ID_OF(VersoPage);

```

```

$PREFIX
$SUFFIX
$NUMBERSTRING
$NUMBER
$STRING
$LogicalObjects
$LogicalObjectClasses

```

)

-- This string expression is allowed in a content generator for CommonReference to automatically generate text for references such as to segment numbers, table numbers, figure numbers, list numbers, footnote numbers and <string> bindings associated with a logical object which is laid out in a current layout object, or to page numbers, pageset numbers and <string> bindings associated with a current or a superior layout object. --

```

DEFINE(COMMONREF, "

```

```

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

```

```

<pre-str> ::= B_REF(SUP(<current-layout-object>))(<prefix>) | ANY_STRING;

```

```

<suf-str> ::= B_REF(SUP(<current-layout-object>))(<suffix>) | ANY_STRING;

```

```

<ref-str> ::= { <ref-numberstring>
    | <ref-fnotestring>
    | <ref-pgnum>
    | <ref-number>
    | <ref-string>
    | ANY_STRING } ... ;

```

```

<ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];

```

```

<pre-str-a> ::= B_REF(SUP(<current-object>))(<prefix>)
    | B_REF(<current-object>)(<prefix>) ;

```

```

<suf-str-a> ::= B_REF(SUP(<current-object>))(<suffix>)
    | B_REF(<current-object>)(<suffix>) ;

```

```

<ref-str-a> ::= B_REF(SUP(<current-object>))(<numberstring>)
    | B_REF(<current-object>)(<numberstring>);

```

```

<ref-fnotestring> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];

```

```

<pre-str-b> ::= B_REF(SUP(<current-logical-object>))(<prefix>)
    | B_REF(<current-logical-object>)(<prefix>) ;

```

```

<suf-str-b> ::= B_REF(SUP(<current-logical-object>))(<suffix>)
    | B_REF(<current-logical-object>)(<suffix>) ;

```

```

<ref-str-b> ::= B_REF(SUP(<current-logical-object>))("fnotestring")
    | B_REF(<current-logical-object>)("fnotestring");

```

```

<ref-pgnum> ::= [<pre-str-c>] <ref-str-c> [<suf-str-c>];

```

```

<pre-str-c> ::= B_REF(SUP(<current-layout-object>))(<prefix>) ;

```

```

<suf-str-c> ::= B_REF(SUP(<current-layout-object>))(<suffix>) ;

```

```

<ref-str-c> ::= MK-STR(B_REF(SUP(<layout-object-1>))("PGnum"))
    | U-ALPHA(B_REF(SUP(<layout-object-1>))("PGnum"))
    | L-ALPHA(B_REF(SUP(<layout-object-1>))("PGnum"))
    | U-ROM(B_REF(SUP(<layout-object-1>))("PGnum"))
    | L-ROM(B_REF(SUP(<layout-object-1>))("PGnum"))

```



```

| MK-STR(B_REF(<layout-object-2>)(""PGnum""))
| U-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
| L-ALPHA(B_REF(<layout-object-2>)(""PGnum""))
| U-ROM(B_REF(<layout-object-2>)(""PGnum""))
| L-ROM(B_REF(<layout-object-2>)(""PGnum""));

<ref-number> ::= [<pre-str-d>] <ref-str-d> [<suf-str-d>];
<pre-str-d> ::= B_REF(SUP(<current-object>))(<prefix>)
| B_REF(<current-object>)(<prefix>);
<suf-str-d> ::= B_REF(SUP(<current-object>))(<suffix>)
| B_REF(<current-object>)(<suffix>);
<ref-str-d> ::= MK-STR(B_REF(SUP(<current-object>))(<number>))
| U-ALPHA(B_REF(SUP(<current-object>))(<number>))
| L-ALPHA(B_REF(SUP(<current-object>))(<number>))
| U-ROM(B_REF(SUP(<current-object>))(<number>))
| L-ROM(B_REF(SUP(<current-object>))(<number>))
| MK-STR(B_REF(<current-object>)(<number>))
| U-ALPHA(B_REF(<current-object>)(<number>))
| L-ALPHA(B_REF(<current-object>)(<number>))
| U-ROM(B_REF(<current-object>)(<number>))
| L-ROM(B_REF(<current-object>)(<number>));

<ref-string> ::= [<pre-str-e>] <ref-str-e> [<suf-str-e>];
<pre-str-e> ::= B_REF(SUP(<current-object>))(<prefix>)
| B_REF(<current-object>)(<prefix>);
<suf-str-e> ::= B_REF(SUP(<current-object>))(<suffix>)
| B_REF(<current-object>)(<suffix>);
<ref-str-e> ::= B_REF(SUP(<current-object>))(<string>)
| B_REF(<current-object>)(<string>);

<current-object> ::= <current-logical-object> | <current-layout-object>;

<current-logical-object> ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
<class-or-type-logical> ::= <logical-object-classes>
| 'composite-logical-object'
| 'basic-logical-object';

<current-layout-object> ::= <layout-object-1> | <layout-object-2>;
<layout-object-1> ::= CURR-INST(<class-or-type-layout-1>,(CURR-OBJ));
<layout-object-2> ::= CURR-INST(<class-or-type-layout-2>,(CURR-OBJ));
<class-or-type-layout-1> ::= 'frame';
<class-or-type-layout-2> ::= 'page'
| OBJECT_CLASS_ID_OF(Page)
| OBJECT_CLASS_ID_OF(RectoPage)
| OBJECT_CLASS_ID_OF(VersoPage);

$PREFIX
$SUFFIX
$NUMBERSTRING
$NUMBER
$STRING
$LogicalObjectClasses
")

```

-- This string expression is allowed in a content generator for CommonNumber to automatically generate text for general references to any kinds of numbers including segment numbers, table numbers, figure numbers, list numbers, footnote numbers, page numbers or pageset numbers, etc. --

```

DEFINE(COMMONNUMBER, "
<string-expr> ::= <ref-numberstring>
| <ref-number>;

<ref-numberstring> ::= [<pre-str-a>] <ref-str-a> [<suf-str-a>];
<pre-str-a> ::= B_REF(SUP(<current-object>))(<prefix>)
| B_REF(<current-object>)(<prefix>) | ANY_STRING;
<suf-str-a> ::= B_REF(SUP(<current-object>))(<suffix>)
| B_REF(<current-object>)(<suffix>) | ANY_STRING;
<ref-str-a> ::= B_REF(SUP(<current-object>))(<numberstring>)
| B_REF(<current-object>)(<numberstring>);

```

```

<ref-number> ::= [<pre-str-b>] <ref-str-b> [<suf-str-b>];
<pre-str-b> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) | ANY_STRING;
<suf-str-b> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) | ANY_STRING;
<ref-str-b> ::= MK-STR(B_REF(SUP(<current-object>))(<number>))
              | U-ALPHA(B_REF(SUP(<current-object>))(<number>))
              | L-ALPHA(B_REF(SUP(<current-object>))(<number>))
              | U-ROM(B_REF(SUP(<current-object>))(<number>))
              | L-ROM(B_REF(SUP(<current-object>))(<number>))
              | MK-STR(B_REF(<current-object>)(<number>))
              | U-ALPHA(B_REF(<current-object>)(<number>))
              | L-ALPHA(B_REF(<current-object>)(<number>))
              | U-ROM(B_REF(<current-object>)(<number>))
              | L-ROM(B_REF(<current-object>)(<number>));

<current-object> ::= <current-logical-object> | <current-layout-object>;

<current-logical-object> ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
<class-or-type-logical> ::= <logical-object-classes>
                          | 'composite-logical-object'
                          | 'basic-logical-object';

<current-layout-object> ::= CURR-INST(<class-or-type-layout>,(CURR-OBJ));
<class-or-type-layout> ::= 'frame'
                          | 'page'
                          | OBJECT_CLASS_ID_OF(Page)
                          | OBJECT_CLASS_ID_OF(RectoPage)
                          | OBJECT_CLASS_ID_OF(VersoPage);

```

```

$PREFIX
$SUFFIX
$NUMBERSTRING
$NUMBER
$LogicalObjectClasses
")

```

-- This string expression is allowed in a content generator for CurrentInstance to automatically generate text for general references to <string> bindings associated with a current logical or layout object. --

```

DEFINE(CURRENTINSTANCE, "
<string-expr> ::= [<pre-str>]<ref-str>[<suf-str>];
<pre-str> ::= B_REF(SUP(<current-object>))(<prefix>)
              | B_REF(<current-object>)(<prefix>) | ANY_STRING;
<suf-str> ::= B_REF(SUP(<current-object>))(<suffix>)
              | B_REF(<current-object>)(<suffix>) | ANY_STRING;
<ref-str> ::= B_REF(SUP(<current-object>))(<string>)
              | B_REF(<current-object>)(<string>);

<current-object> ::= <current-logical-object> | <current-layout-object>;

<current-logical-object> ::= CURR-INST(<class-or-type-logical>,(CURR-OBJ));
<class-or-type-logical> ::= <logical-object-classes>
                          | 'composite-logical-object'
                          | 'basic-logical-object';

<current-layout-object> ::= CURR-INST(<class-or-type-layout>,(CURR-OBJ));
<class-or-type-layout> ::= 'frame'
                          | 'page'
                          | OBJECT_CLASS_ID_OF(Page)
                          | OBJECT_CLASS_ID_OF(RectoPage)
                          | OBJECT_CLASS_ID_OF(VersoPage);

```

```

$PREFIX
$SUFFIX
$STRING
$LogicalObjectClasses
")

```

-- This string expression is allowed in a content generator for GenericBlock to automatically generate text for general references to bindings associated with a current layout object. --

```

DEFINE(GENERICBLOCKREF, "
<string-expr> ::= [<pre-str>]<ref-str>[<suf-str>];

<pre-str> ::= B_REF(SUP(CURR-OBJ))(<prefix>) | ANY_STRING;
<suf-str> ::= B_REF(SUP(CURR-OBJ))(<suffix>) | ANY_STRING;
<ref-str> ::= MK-STR(B_REF(SUP(CURR-OBJ))(<number>))
| U-ALPHA(B_REF(SUP(CURR-OBJ))(<number>))
| L-ALPHA(B_REF(SUP(CURR-OBJ))(<number>))
| U-ROM(B_REF(SUP(CURR-OBJ))(<number>))
| L-ROM(B_REF(SUP(CURR-OBJ))(<number>))
| MK-STR(B_REF(SUP(CURR-OBJ))("PGnum"))
| U-ALPHA(B_REF(SUP(CURR-OBJ))("PGnum"))
| L-ALPHA(B_REF(SUP(CURR-OBJ))("PGnum"))
| U-ROM(B_REF(SUP(CURR-OBJ))("PGnum"))
| L-ROM(B_REF(SUP(CURR-OBJ))("PGnum"))
| B_REF(SUP(CURR-OBJ))(<numberstring>)
| B_REF(SUP(CURR-OBJ))(<string>);

```

```

$PREFIX
$SUFFIX
$NUMBER
$NUMBERSTRING
$STRING
")

```

```

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

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

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

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

```

```

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

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

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

<construction-factor> ::= OBJECT_CLASS_ID_OF(Passage)
| OBJECT_CLASS_ID_OF(NumberedSegment)
| OBJECT_CLASS_ID_OF(Paragraph)
| OBJECT_CLASS_ID_OF(BodyText)
| OBJECT_CLASS_ID_OF(BodyRaster)
| OBJECT_CLASS_ID_OF(BodyGeometric)
| OBJECT_CLASS_ID_OF(Figure)

```

```

| OBJECT_CLASS_ID_OF(Table)
| OBJECT_CLASS_ID_OF(NumberedList)
| OBJECT_CLASS_ID_OF(UnNumberedList)
| OBJECT_CLASS_ID_OF(DefinitionList)
| <construction-type>;
")
DEFINE(CONSTRAINT-2, "
<constraint-2> ::= OBJECT_CLASS_ID_OF(Title)
| OPT OBJECT_CLASS_ID_OF(Title);
")
DEFINE(PassageGFS, "
<construction-expr> ::= <constraint-1>
| SEQ(<constraint-2><constraint-1>);

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

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

$CONSTRAINT-1
$CONSTRAINT-2
")
DEFINE(CONSTRAINT-3, "
<construction-expr> ::= <construction-term>
| <construction-type>;

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

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

<construction-factor> ::= OBJECT_CLASS_ID_OF(BodyText)
| OBJECT_CLASS_ID_OF(BodyRaster)
| OBJECT_CLASS_ID_OF(BodyGeometric)
| OBJECT_CLASS_ID_OF(Footnote)
| OBJECT_CLASS_ID_OF(Phrase)
| OBJECT_CLASS_ID_OF(Reference)
| <construction-type>;
")
DEFINE(ParagraphGFS, "$CONSTRAINT-3")
DEFINE(TitleGFS, "$CONSTRAINT-3")
DEFINE(FigureGFS, "
<construction-expr> ::=
SEQ([<term-1>][<term-2>][<term-3>]<term-4>)
| SEQ([<term-1>][<term-2>]<term-4>[<term-3>])
| SEQ([<term-3>][<term-1>][<term-2>]<term-4>)
| SEQ(<term-4>[<term-1>][<term-2>][<term-3>])
| SEQ([<term-3>]<term-4>[<term-1>][<term-2>])
| SEQ(<term-4>[<term-3>][<term-1>][<term-2>]);

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

<term-2> ::= OBJECT_CLASS_ID_OF(Caption)
| OPT OBJECT_CLASS_ID_OF(Caption);

<term-3> ::= OBJECT_CLASS_ID_OF(Description)
| OPT OBJECT_CLASS_ID_OF(Description);

<term-4> ::= OBJECT_CLASS_ID_OF(Artwork)
| OBJECT_CLASS_ID_OF(Form);
")

```

```

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

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

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

<construction-factor> ::= OBJECT_CLASS_ID_OF(Phrase)
                        | OBJECT_CLASS_ID_OF(BodyRaster)
                        | OBJECT_CLASS_ID_OF(BodyGeometric)
                        | <construction-type>;

")

DEFINE(CONSTRAINT-5, "
<construction-expr> ::= <construction-term>
                        | <construction-type>;

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

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

<construction-factor> ::= OBJECT_CLASS_ID_OF(Phrase)
                        | OBJECT_CLASS_ID_OF(Footnote)
                        | OBJECT_CLASS_ID_OF(Reference)
                        | OBJECT_CLASS_ID_OF(BodyText)
                        | <construction-type>;

")

DEFINE(PhraseGFS, "$CONSTRAINT-5")
DEFINE(CaptionGFS, "$CONSTRAINT-5")
DEFINE(DescriptionGFS, "$CONSTRAINT-5")

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

")

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

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

")

DEFINE(ReferenceGFS, "
<construction-expr> ::= OBJECT_CLASS_ID_OF(ReferencedContent)
                        | SEQ([<term>]
                            OBJECT_CLASS_ID_OF(ReferencedContent)
                            [<term>]);

<term> ::= OBJECT_CLASS_ID_OF(BodyText)
          | OPT OBJECT_CLASS_ID_OF(BodyText)
          | CHO( {OBJECT_CLASS_ID_OF(BodyText)}... );

")

```

```

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

<construction-factor> ::= OBJECT_CLASS_ID_OF(CommonText)
                        | OBJECT_CLASS_ID_OF(PageNumber)
                        | OBJECT_CLASS_ID_OF(CommonRaster)
                        | OBJECT_CLASS_ID_OF(CommonGeometric)
                        | OBJECT_CLASS_ID_OF(CommonReference)
                        | OBJECT_CLASS_ID_OF(CommonNumber)
                        | OBJECT_CLASS_ID_OF(CurrentInstance)
                        | OBJECT_CLASS_ID_OF(TableNumber);

")

DEFINE(TableGFS, "
<construction-expr> ::= REP CHO(OBJECT_CLASS_ID_OF(Row)...)
                        | REP OBJECT_CLASS_ID_OF(Row)
                        | SEQ(OBJECT_CLASS_ID_OF(Row)...);

")

DEFINE(RowGFS, "
<construction-expr> ::= <simple-table> | <complex-table>;
<simple-table> ::= REP OBJECT_CLASS_ID_OF(EntryElement)
                  | REP CHO(OBJECT_CLASS_ID_OF(EntryElement)...)
                  | SEQ(OBJECT_CLASS_ID_OF(EntryElement)...);
<complex-table> ::= SEQ(OBJECT_CLASS_ID_OF(EntryElement)
                       OBJECT_CLASS_ID_OF(TableComponent));

")

DEFINE(TableComponentGFS, "
<construction-expr> ::= REP OBJECT_CLASS_ID_OF(RowComponent);

")

DEFINE(RowComponentGFS, "
<construction-expr> ::= REP OBJECT_CLASS_ID_OF(EntryElement)
                        | REP CHO(OBJECT_CLASS_ID_OF(EntryElement)...)
                        | SEQ(OBJECT_CLASS_ID_OF(EntryElement)...);

")

DEFINE(FormGFS, "
<construction-expr> ::= AGG(<factor>...);
<factor> ::= OBJECT_CLASS_ID_OF(EntryElement)
           | OBJECT_CLASS_ID_OF(EntryGroup);

")

DEFINE(EntryGroupGFS, "$FormGFS")

DEFINE(EntryElementGFS, "
<construction-expr> ::= OBJECT_CLASS_ID_OF(EntryText)
                        | OBJECT_CLASS_ID_OF(EntryRaster)
                        | OBJECT_CLASS_ID_OF(EntryGeometric);

")

DEFINE(NumberedListGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(Number)
                           OBJECT_CLASS_ID_OF(ListItem))
                        | REP SEQ(OBJECT_CLASS_ID_OF(Number)
                                  OBJECT_CLASS_ID_OF(ListItem));

")

DEFINE(UnNumberedListGFS, "
<construction-expr> ::= OBJECT_CLASS_ID_OF(ListItem)
                        | REP OBJECT_CLASS_ID_OF(ListItem)
                        | SEQ(<separator-obj> OBJECT_CLASS_ID_OF(ListItem))
                        | REP SEQ(<separator-obj> OBJECT_CLASS_ID_OF(ListItem));

<separator-obj> ::= OBJECT_CLASS_ID_OF(BodyText)
                 | OBJECT_CLASS_ID_OF(BodyRaster)
                 | OBJECT_CLASS_ID_OF(BodyGeometric);

")

```

```

DEFINE(DefinitionListGFS,"
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(ListTerm)
OBJECT_CLASS_ID_OF(ListItem))
|REP SEQ(OBJECT_CLASS_ID_OF(ListTerm)
OBJECT_CLASS_ID_OF(ListItem));
")

```

```

DEFINE(ListItemGFS,"
<construction-expr> ::= <term> | CHO(<term>...);

<term> ::= REP OBJECT_CLASS_ID_OF(Phrase)
| OBJECT_CLASS_ID_OF(NumberedList)
| OBJECT_CLASS_ID_OF(UnNumberedList)
| OBJECT_CLASS_ID_OF(DefinitionList);
")

```

```

DEFINE(ListTermGFS,"$CONSTRAINT-3")

```

7.3.2 Factor constraints

7.3.2.1 FACTOR ANY-LOGICAL

```

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

```

7.3.2.2 FACTOR COMP-LOGICAL

```

:ANY-LOGICAL {
GENERIC:
    REQObject-type           {'composite-logical-object'}
SPECIFIC:
    REQSubordinates          {VIRTUAL},
    PERM Object-type         {'composite-logical-object'}
SPECIFIC_AND_GENERIC:
    PERM Bindings            {PMUL {$SPECIFYBINDINGS}},
    PERM Default-value-lists {REQ #basic-logical-attributes
                                {PERM #presentation-style {ANY_VALUE},
                                PERM #layout-style {ANY_VALUE}}}
}

```

7.3.2.3 FACTOR BASIC-LOGICAL

```

:ANY-LOGICAL {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM Resource             {ANY_VALUE}
SPECIFIC:
    PERM Object-type         {'basic-logical-object'}
SPECIFIC_AND_GENERIC:
    PERM Bindings            {PMUL {$SPECIFYBINDINGS}}
}

```

7.3.2.4 FACTOR ANY-COMMON

```
{  
GENERIC:  
  REQObject-type           {VIRTUAL},  
  REQObject-class-identifier {ANY_VALUE},  
  PERM Bindings             {PMUL {$SPECIFYBINDINGS}},  
  PERM Protection          {ANY_VALUE},  
  PERM User-readable-comments {ANY_STRING},  
  PERM User-visible-name    {ANY_STRING}  
}
```

7.3.3 Constituent constraints

7.3.3.1 DocumentLogicalRoot

:ANY-LOGICAL {

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

SPECIFIC:

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

SPECIFIC_AND_GENERIC:

```
  PERM Bindings             {PMUL {$SPECIFYBINDINGS},  
                             PERM {$INITIALISEFNOTE}},  
  PERM Default-value-lists  {REQ #basic-logical-attributes  
                             {PERM #presentation-style {ANY_VALUE},  
                             PERM #layout-style {ANY_VALUE}}
```

}

7.3.3.2 Passage

:COMP-LOGICAL {

```
GENERIC:  
  REQGenerator-for-subordinates {$PassageGFS},  
  REQApplication-comments    {REQ #constraint-name {"1"},  
                               PERM #external-data {ANY_VALUE}}
```

SPECIFIC:

```
  REQObject-class           {OBJECT_CLASS_ID_OF(Passage)},  
  REQSubordinates           {SUB_ID_OF(Title),  
                             SUB_ID_OF(Passage)+,  
                             SUB_ID_OF(NumberedSegment)+,  
                             SUB_ID_OF(BodyText)+,  
                             SUB_ID_OF(BodyRaster)+,  
                             SUB_ID_OF(BodyGeometric)+,  
                             SUB_ID_OF(Figure)+,  
                             SUB_ID_OF(Paragraph)+,  
                             SUB_ID_OF(Table)+,  
                             SUB_ID_OF(NumberedList)+,  
                             SUB_ID_OF(DefinitionList)+,  
                             SUB_ID_OF(UnNumberedList)+,  
  PERM Application-comments  {REQ #constraint-name {"1"},  
                               PERM #external-data {ANY_VALUE}}
```

SPECIFIC_AND_GENERIC:

```
  PERM Bindings             {PMUL {$SPECIFYBINDINGS},  
                             PERM {$INITIALISEFNOTE}},  
  PERM Layout-style         {STYLE_ID_OF(LStyle1)}
```

}

7.3.3.3 NumberedSegment

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$NumberedSegmentGFS},
REQApplication-comments {REQ #constraint-name {"2"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(NumberedSegment)},
REQSubordinates {SUB_ID_OF(Number),
SUB_ID_OF(Title),
SUB_ID_OF(Passage)+,
SUB_ID_OF(NumberedSegment)+,
SUB_ID_OF(BodyText)+,
SUB_ID_OF(BodyRaster)+,
SUB_ID_OF(BodyGeometric)+,
SUB_ID_OF(Paragraph)+,
SUB_ID_OF(Figure)+,
SUB_ID_OF(Table)+,
SUB_ID_OF(NumberedList)+,
SUB_ID_OF(DefinitionList)+,
SUB_ID_OF(UnNumberedList)+},
PERM Application-comments {REQ #constraint-name {"2"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.4 Number

:BASIC-LOGICAL {

GENERIC:

REQContent-generator {\$SEGMENTNUMBER},
REQApplication-comments {REQ #constraint-name {"3"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Number)},
PERM Content-generator {\$SEGMENTNUMBER},
PERM Application-comments {REQ #constraint-name {"3"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle2)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC\$PC\$FPC}

}

7.3.3.5 Title

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$TitleGFS},
REQApplication-comments {REQ #constraint-name {"4"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Title)},
REQSubordinates {SUB_ID_OF(BodyText)+,
SUB_ID_OF(BodyRaster)+,
SUB_ID_OF(BodyGeometric)+,
SUB_ID_OF(Phrase)+,
SUB_ID_OF(Footnote)+,
SUB_ID_OF(Reference)+},
PERM Application-comments {REQ #constraint-name {"4"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.6 Caption

```
:COMP-LOGICAL {  
GENERIC:  
    REQGenerator-for-subordinates    {$CaptionGFS},  
    REQApplication-comments          {REQ #constraint-name {"5"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC:  
    REQObject-class                  {OBJECT_CLASS_ID_OF(Caption)},  
    REQSubordinates                  {SUB_ID_OF(BodyText)+,  
                                     SUB_ID_OF(Phrase)+,  
                                     SUB_ID_OF(Footnote)+,  
                                     SUB_ID_OF(Reference)+},  
    PERM Application-comments        {REQ #constraint-name {"5"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC_AND_GENERIC:  
    PERM Layout-style                {STYLE_ID_OF(LStyle1)}  
}
```

7.3.3.7 Paragraph

```
:COMP-LOGICAL {  
GENERIC:  
    REQGenerator-for-subordinates    {$ParagraphGFS},  
    REQApplication-comments          {REQ #constraint-name {"6"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC:  
    REQObject-class                  {OBJECT_CLASS_ID_OF(Paragraph)},  
    REQSubordinates                  {SUB_ID_OF(BodyText)+,  
                                     SUB_ID_OF(Footnote)+,  
                                     SUB_ID_OF(BodyRaster)+,  
                                     SUB_ID_OF(BodyGeometric)+,  
                                     SUB_ID_OF(Phrase)+,  
                                     SUB_ID_OF(Reference)+},  
    PERM Application-comments        {REQ #constraint-name {"6"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC_AND_GENERIC:  
    PERM Layout-style                {STYLE_ID_OF(LStyle1)}  
}
```

7.3.3.8 Phrase

```
:COMP-LOGICAL {  
GENERIC:  
    REQGenerator-for-subordinates    {$PhraseGFS},  
    REQApplication-comments          {REQ #constraint-name {"7"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC:  
    REQObject-class                  {OBJECT_CLASS_ID_OF(Phrase)},  
    REQSubordinates                  {SUB_ID_OF(BodyText)+,  
                                     SUB_ID_OF(Footnote)+,  
                                     SUB_ID_OF(Phrase)+,  
                                     SUB_ID_OF(Reference)+},  
    PERM Application-comments        {REQ #constraint-name {"7"},  
                                     PERM #external-data {ANY_VALUE}}  
SPECIFIC_AND_GENERIC:  
    PERM Layout-style                {STYLE_ID_OF(LStyle1)}  
}
```

7.3.3.9 Footnote

```
:COMP-LOGICAL {  
GENERIC:  
    REQGenerator-for-subordinates    {$FootnoteGFS},  
    PERM Bindings                    {PMUL {$SPECIFYBINDINGS},  
                                     PERM {REQ{$INCFNOTENUMBER,$FNOTENUMBERSTRING}  
                                     | $FNOTESTRINGLITERAL}},  
    REQApplication-comments          {REQ #constraint-name {"8"},  
                                     PERM #external-data {ANY_VALUE}}  
}
```

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Footnote)},
REQSubordinates {SUB_ID_OF(FootnoteReference),
SUB_ID_OF(FootnoteBody)},
PERM Bindings {PMUL {\$SPECIFYBINDINGS},
PERM {\$FNOTESTRINGLITERAL}},
PERM Application-comments {REQ #constraint-name {"8"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.10 FootnoteNumber

:BASIC-LOGICAL {

GENERIC:

REQContent-generator {\$FNNUMBER},
REQApplication-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(FootnoteNumber)},
PERM Content-generator {\$FNNUMBER},
PERM Application-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle9)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC|PC|FPC}

}

7.3.3.11 FootnoteReference

:BASIC-LOGICAL {

GENERIC:

REQContent-generator {\$FNNUMBER},
REQApplication-comments {REQ #constraint-name {"10"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(FootnoteReference)},
PERM Content-generator {\$FNNUMBER},
PERM Application-comments {REQ #constraint-name {"10"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle10)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC|PC|FPC}

}

7.3.3.12 FootnoteBody

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$FootnoteBodyGFS},
REQApplication-comments {REQ #constraint-name {"11"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(FootnoteBody)},
REQSubordinates {SUB_ID_OF(FootnoteNumber),
SUB_ID_OF(FootnoteText)+,
SUB_ID_OF(Reference)+},
PERM Application-comments {REQ #constraint-name {"11"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle11)}

}

7.3.3.13 FootnoteText

:BASIC-LOGICAL {

GENERIC:

REQApplication-comments {REQ #constraint-name {"12"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(FootnoteText)},
PERM Application-comments {REQ #constraint-name {"12"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle6)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC\$PC\$FPC},

PERM Content-portions {CONTENT_ID_OF(Character-content-portion)+}

}

7.3.3.14 Figure

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$FigureGFS},
REQApplication-comments {REQ #constraint-name {"13"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Figure)},
REQSubordinates {SUB_ID_OF(Number),
SUB_ID_OF(Caption),
SUB_ID_OF(Description),
SUB_ID_OF(Artwork),
SUB_ID_OF(Form)},

PERM Application-comments {REQ #constraint-name {"13"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.15 BodyText

:BASIC-LOGICAL {

GENERIC:

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

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(BodyText)},
PERM Application-comments {REQ #constraint-name {"14"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle2)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC\$PC\$FPC},
PERM Content-portions {CONTENT_ID_OF(Character-content-portion)+}

-- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the attribute "resource" shall be specified. --

}

7.3.3.16 Reference

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$ReferenceGFS},
REQApplication-comments {REQ #constraint-name {"15"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Reference)},
REQSubordinates {SUB_ID_OF(BodyText)+,
SUB_ID_OF(ReferencedContent)},
PERM Application-comments {REQ #constraint-name {"15"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.17 ReferencedContent

:BASIC-LOGICAL {

GENERIC:

REQApplication-comments {REQ #constraint-name {"16"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(ReferencedContent)},
PERM Content-generator {\$REF},
PERM Content-portions {CONTENT_ID_OF(Character-content-portion)+,
-- Either Content-generator or Content-portions is specified. --
PERM Application-comments {REQ #constraint-name {"16"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle10)},
PERM Presentation-style {STYLE_ID_OF(PStyle1)},
PERM Content-architecture-class {\$FC|\$PC|\$FPC}

}

7.3.3.18 BodyRaster

:BASIC-LOGICAL {

GENERIC:

REQContent-architecture-class {\$FPR},
REQApplication-comments {REQ #constraint-name {"17"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(BodyRaster)},
PERM Content-architecture-class {\$FPR},
PERM Application-comments {REQ #constraint-name {"17"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle5)},
PERM Presentation-style {STYLE_ID_OF(PStyle3)},
PERM Content-portions {CONTENT_ID_OF(Raster-graphics-content-portion)}

-- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the attribute "resource" shall be specified. --

}

7.3.3.19 BodyGeometric

:BASIC-LOGICAL {

GENERIC:

REQContent-architecture-class {\$FPG},
REQApplication-comments {REQ #constraint-name {"18"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(BodyGeometric)},
PERM Content-architecture-class {\$FPG},
PERM Application-comments {REQ #constraint-name {"18"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle5)},
PERM Presentation-style {STYLE_ID_OF(PStyle2)},
PERM Content-portions {CONTENT_ID_OF(Geometric-graphics-content-portion)}

-- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the attribute "resource" shall be specified. --

}

7.3.3.20 CommonContent

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'composite-logical-object'},
    REQGenerator-for-subordinates {$CommonContentGFS},
    REQApplication-comments    {REQ #constraint-name {"19"},
                                PERM #external-data {ANY_VALUE}},
    PERM   Default-value-lists {REQ #basic-logical-attributes
                                {PERM #presentation-style      {ANY_VALUE},
                                 PERM #layout-style             {ANY_VALUE}}}
}
```

7.3.3.21 CommonText

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM   Content-portions  {CONTENT_ID_OF(Character-content-portion)+},
    PERM   Resource          {ANY_VALUE},
    PERM   Layout-style      {STYLE_ID_OF(LStyle3)},
    PERM   Presentation-style {STYLE_ID_OF(PStyle4)},
    PERM   Content-architecture-class {$FC$PC$FPC},
    REQApplication-comments    {REQ #constraint-name {"20"},
                                PERM #external-data {ANY_VALUE}}
    -- Either the attribute "content portions" or "resource" shall be specified in the above constituent
    constraint. --
}
```

7.3.3.22 CommonReference

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

7.3.3.23 CommonNumber

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM   Content-generator  {$COMMONNUMBER},
    PERM   Layout-style       {STYLE_ID_OF(LStyle3)},
    PERM   Presentation-style  {STYLE_ID_OF(PStyle4)},
    PERM   Content-architecture-class {$FC$PC$FPC},
    REQApplication-comments    {REQ #constraint-name {"38"},
                                PERM #external-data {ANY_VALUE}}
}
```

7.3.3.24 CurrentInstance

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM   Content-generator  {$CURRENTINSTANCE},
    PERM   Layout-style       {STYLE_ID_OF(LStyle3)},
    PERM   Presentation-style  {STYLE_ID_OF(PStyle4)},
    PERM   Content-architecture-class {$FC$PC$FPC},
    REQApplication-comments    {REQ #constraint-name {"39"},
                                PERM #external-data {ANY_VALUE}}
}
```

7.3.3.25 CommonRaster

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM Content-portions    {CONTENT_ID_OF(Raster-graphics-content-portion)},
    PERM Resource             {ANY_VALUE},
    PERM Layout-style        {STYLE_ID_OF(LStyle8)},
    PERM Presentation-style   {STYLE_ID_OF(PStyle3)},
    REQContent-architecture-class {$FPR},
    REQApplication-comments    {REQ #constraint-name {"21"},
                                PERM #external-data {ANY_VALUE}}
    -- Either the attribute "content portions" or "resource" shall be specified in the above constituent
    constraint. --
}
```

7.3.3.26 CommonGeometric

```
:ANY-COMMON {
GENERIC:
    REQObject-type           {'basic-logical-object'},
    PERM Content-portions    {CONTENT_ID_OF(Geometric-graphics-content-portion)},
    PERM Resource             {ANY_VALUE},
    PERM Layout-style        {STYLE_ID_OF(LStyle8)},
    PERM Presentation-style   {STYLE_ID_OF(PStyle2)},
    REQContent-architecture-class {$FPG},
    REQApplication-comments    {REQ #constraint-name {"22"},
                                PERM #external-data {ANY_VALUE}}
    -- Either the attribute "content portions" or "resource" shall be specified in the above constituent
    constraint --
}
```

7.3.3.27 PageNumber

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

7.3.3.28 TableNumber

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

7.3.3.29 Description

```
:COMP-LOGICAL {
GENERIC:
    REQGenerator-for-subordinates {$DescriptionGFS},
    REQApplication-comments    {REQ #constraint-name {"23"},
                                PERM #external-data {ANY_VALUE}}
}
```

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Description)},
REQSubordinates {SUB_ID_OF(BodyText)+,
SUB_ID_OF(Footnote)+,
SUB_ID_OF(Phrase)+,
SUB_ID_OF(Reference)+},
PERM Application-comments {REQ #constraint-name {"23"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.30 Artwork

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$ArtworkGFS},
REQApplication-comments {REQ #constraint-name {"24"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(Artwork)},
REQSubordinates {SUB_ID_OF(BodyRaster)+,
SUB_ID_OF(BodyGeometric)+,
SUB_ID_OF(Phrase)+},
PERM Application-comments {REQ #constraint-name {"24"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle12)}

}

7.3.3.31 NumberedList

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$NumberedListGFS},
REQApplication-comments {REQ #constraint-name {"25"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(NumberedList)},
REQSubordinates {SUB_ID_OF(Number)+,
SUB_ID_OF(ListItem)+},
PERM Application-comments {REQ #constraint-name {"25"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.32 UnNumberedList

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$UnNumberedListGFS},
REQApplication-comments {REQ #constraint-name {"26"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(UnNumberedList)},
REQSubordinates {SUB_ID_OF(BodyText)+,
SUB_ID_OF(BodyRaster)+,
SUB_ID_OF(BodyGeometric)+,
SUB_ID_OF(ListItem)+},
PERM Application-comments {REQ #constraint-name {"26"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.33 DefinitionList

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$DefinitionListGFS},
REQApplication-comments {REQ #constraint-name {"27"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(DefinitionList)},
REQSubordinates {SUB_ID_OF(ListTerm)+,
SUB_ID_OF(ListItem)+},
PERM Application-comments {REQ #constraint-name {"27"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.34 ListItem

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$ListItemGFS},
REQApplication-comments {REQ #constraint-name {"28"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(ListItem)},
REQSubordinates {SUB_ID_OF(Phrase)+,
SUB_ID_OF(NumberedList)+,
SUB_ID_OF(UnNumberedList)+,
SUB_ID_OF(DefinitionList)+},
PERM Application-comments {REQ #constraint-name {"28"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.35 ListTerm

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$ListTermGFS},
REQApplication-comments {REQ #constraint-name {"29"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

REQObject-class {OBJECT_CLASS_ID_OF(ListTerm)},
REQSubordinates {SUB_ID_OF(BodyText)+,
SUB_ID_OF(BodyRaster)+,
SUB_ID_OF(BodyGeometric)+,
SUB_ID_OF(Reference)+,
SUB_ID_OF(Phrase)+,
SUB_ID_OF(Footnote)+},
PERM Application-comments {REQ #constraint-name {"29"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:

PERM Layout-style {STYLE_ID_OF(LStyle1)}

}

7.3.3.36 Table

:COMP-LOGICAL {

GENERIC:

REQGenerator-for-subordinates {\$TableGFS},
REQApplication-comments {REQ #constraint-name {"30"},
PERM #external-data {ANY_VALUE}},
REQLayout-style {STYLE_ID_OF(LStyleT4)}

SPECIFIC:

```
REQObject-class      {OBJECT_CLASS_ID_OF(Table)},
REQSubordinates      {SUB_ID_OF(Row)+},
PERM  Application-comments  {REQ #constraint-name {"30"},
                           PERM #external-data {ANY_VALUE}},
PERM  Layout-style      {STYLE_ID_OF(LStyleT8)}
```

}

7.3.3.37 Row

:COMP-LOGICAL {

GENERIC:

```
REQGenerator-for-subordinates  {$RowGFS},
REQApplication-comments        {REQ #constraint-name {"31"},
                               PERM #external-data {ANY_VALUE}},
REQLayout-style                {STYLE_ID_OF(LStyleT5)}
```

SPECIFIC:

```
REQObject-class      {OBJECT_CLASS_ID_OF(Row)},
REQSubordinates      {SUB_ID_OF(EntryElement)+,
                     SUB_ID_OF(TableComponent)},
PERM  Application-comments  {REQ #constraint-name {"31"},
                           PERM #external-data {ANY_VALUE}}
```

}

7.3.3.38 TableComponent

:COMP-LOGICAL {

GENERIC:

```
REQGenerator-for-subordinates  {$TableComponentGFS},
REQApplication-comments        {REQ #constraint-name {"32"},
                               PERM #external-data {ANY_VALUE}},
REQLayout-style                {STYLE_ID_OF(LStyleT6)}
```

SPECIFIC:

```
REQObject-class      {OBJECT_CLASS_ID_OF(TableComponent)},
REQSubordinates      {SUB_ID_OF(RowComponent)+},
PERM  Application-comments  {REQ #constraint-name {"32"},
                           PERM #external-data {ANY_VALUE}}
```

}

7.3.3.39 RowComponent

:COMP-LOGICAL {

GENERIC:

```
REQGenerator-for-subordinates  {$RowComponentGFS},
REQApplication-comments        {REQ #constraint-name {"33"},
                               PERM #external-data {ANY_VALUE}},
REQ  Layout-style              {STYLE_ID_OF(LStyleT7)}
```

SPECIFIC:

```
REQObject-class      {OBJECT_CLASS_ID_OF(RowComponent)},
REQSubordinates      {SUB_ID_OF(EntryElement)+,
                     SUB_ID_OF(EntryGroup)+},
PERM  Application-comments  {REQ #constraint-name {"33"},
                           PERM #external-data {ANY_VALUE}}
```

}

7.3.3.40 Form

:COMP-LOGICAL {

GENERIC:

```
REQGenerator-for-subordinates  {$FormGFS},
REQApplication-comments        {REQ #constraint-name {"34"},
                               PERM #external-data {ANY_VALUE}},
REQLayout-style                {STYLE_ID_OF(LStyleT1)}
```

SPECIFIC:

```
REQObject-class      {OBJECT_CLASS_ID_OF(Form)},
REQSubordinates      {SUB_ID_OF(EntryElement)+,
                     SUB_ID_OF(EntryGroup)+},
PERM  Application-comments  {REQ #constraint-name {"34"},
                           PERM #external-data {ANY_VALUE}}
```

}

7.3.3.41 EntryElement

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

7.3.3.42 EntryGroup

```
:COMP-LOGICAL {
GENERIC:
    REQGenerator-for-subordinates    {$EntryGroupGFS},
    REQApplication-comments          {REQ #constraint-name {"36"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
    REQObject-class                  {OBJECT_CLASS_ID_OF(EntryGroup)},
    REQSubordinates                  {SUB_ID_OF(EntryElement)+,
                                     SUB_ID_OF(EntryGroup)+},
    PERM Application-comments        {REQ #constraint-name {"36"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Layout-style                {STYLE_ID_OF(LStyleT3)}
}
```

7.3.3.43 EntryText

```
:BASIC-LOGICAL {
GENERIC:
    REQApplication-comments          {REQ #constraint-name {"41"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
    REQObject-class                  {OBJECT_CLASS_ID_OF(EntryText)},
    PERM Application-comments        {REQ #constraint-name {"41"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Layout-style                {STYLE_ID_OF(LStyleT9)},
    PERM Presentation-style          {STYLE_ID_OF(PStyle1)},
    PERM Content-architecture-class  {$FC$PC$FPC},
    PERM Content-portions            {CONTENT_ID_OF(Character-content-portion)+}
    -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
    attribute "resource" shall be specified. --
}
```

7.3.3.44 EntryRaster

```
:BASIC-LOGICAL {
GENERIC:
    REQContent-architecture-class    {$FPR},
    REQApplication-comments          {REQ #constraint-name {"42"},
                                     PERM #external-data {ANY_VALUE}}
SPECIFIC:
    REQObject-class                  {OBJECT_CLASS_ID_OF(EntryRaster)},
    PERM Content-architecture-class  {$FPR},
    PERM Application-comments        {REQ #constraint-name {"42"},
                                     PERM #external-data {ANY_VALUE}}
}
```

SPECIFIC_AND_GENERIC:

```
    PERM  Layout-style           {STYLE_ID_OF(LStyleT9)},
    PERM  Presentation-style     {STYLE_ID_OF(PStyle3)},
    PERM  Content-portions       {CONTENT_ID_OF(Raster-graphics-content-portion)}
    -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
    attribute "resource" shall be specified. --
```

}

7.3.3.45 EntryGeometric

:BASIC-LOGICAL {

GENERIC:

```
    REQContent-architecture-class  {$FPG},
    REQApplication-comments         {REQ #constraint-name {"43"},
                                     PERM #external-data {ANY_VALUE}}
```

SPECIFIC:

```
    REQObject-class                {OBJECT_CLASS_ID_OF(EntryGeometric)},
    PERM  Content-architecture-class  {$FPG},
    PERM  Application-comments        {REQ #constraint-name {"43"},
                                     PERM #external-data {ANY_VALUE}}
```

SPECIFIC_AND_GENERIC:

```
    PERM  Layout-style           {STYLE_ID_OF(LStyleT9)},
    PERM  Presentation-style     {STYLE_ID_OF(PStyle2)},
    PERM  Content-portions       {CONTENT_ID_OF(Geometric-graphics-content-portion)}
    -- If the attribute "content portions" is specified neither in the specific nor in the generic part, then the
    attribute "resource" shall be specified. --
```

}

7.4 Layout constituent constraints

7.4.1 Macro definitions

DEFINE(DocLayRootGFS, "

```
<construction-expr> ::= <construction-term> | <construction-type>;
```

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

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

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

")

DEFINE(PageSetGFS, "

```
<construction-expr> ::= <construction-term> | <construction-type>;
```

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

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

```
<construction-factor> ::= OBJECT_CLASS_ID_OF(Page)
                        | OBJECT_CLASS_ID_OF(RectoPage)
                        | OBJECT_CLASS_ID_OF(VersoPage)
                        | <construction-type>;
```

")

DEFINE(PageGFS, "

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

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

```

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

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

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

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

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

DEFINE(HeaderFooterGFS, "$CompositeCommonGFS")

DEFINE(FixedCompositeBodyGFS, "
<construction-expr> ::= SEQ(<construction-term>...);

<construction-term> ::= <construction-factor1>
                    | OPT <construction-factor1>
                    | CHO({<construction-factor1>}...)
                    | <construction-factor2>;

<construction-factor1> ::= OBJECT_CLASS_ID_OF(BasicFixture)
                        | OBJECT_CLASS_ID_OF(ColumnFixed)
                        | OBJECT_CLASS_ID_OF(CompositeFixtureFixed)
                        | OBJECT_CLASS_ID_OF(VariableCompositeBody);

<construction-factor2> ::= OBJECT_CLASS_ID_OF(CompositeCommon)
                        | OBJECT_CLASS_ID_OF(SourcedContentFixed)
                        | OBJECT_CLASS_ID_OF(ArrangedContentFixed);
")

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

<construction-term> ::= <construction-factor1>
                    | OPT <construction-factor1>
                    | REP <construction-factor1>
                    | OPT REP <construction-factor1>;

<construction-type> ::= SEQ({<construction-term>|<construction-factor2>}...)
                    | CHO({<construction-term>}...);

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

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

<construction-factor2> ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                        | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")

```

```

DEFINE(SnakingColumnsGFS, "
<construction-expr> ::= REP <construction-factor1>
                        | <construction-term>
                        | SEQ(<construction-type>...);
<construction-term> ::= SEQ(<construction-type><following-term>);
<following-term> ::= OPT<construction-factor1>
                    | <construction-factor2>
                    | OPT<construction-term> ;

<construction-type> ::= <construction-factor1>|<construction-factor2>;
<construction-factor1> ::= OBJECT_CLASS_ID_OF(ColumnVariable)
                        | OBJECT_CLASS_ID_OF(CompositeColumnVariable);
<construction-factor2> ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                        | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")

DEFINE(SynchronizedColumnsGFS, "
<construction-expr> ::= SEQ({<construction-type>}...)|<construction-term>;
<construction-term> ::= SEQ(<construction-type><following-term>);
<following-term> ::= OPT<construction-factor1>
                    | <construction-factor2>
                    | OPT<construction-term> ;

<construction-type> ::= <construction-factor1>|<construction-factor2>;
<construction-factor1> ::= OBJECT_CLASS_ID_OF(ColumnFixed)
                        | OBJECT_CLASS_ID_OF(CompositeColumnFixed);
<construction-factor2> ::= OBJECT_CLASS_ID_OF(ArrangedContentFixed)
                        | OBJECT_CLASS_ID_OF(SourcedContentFixed);
")

DEFINE(CompositeFloatGFS, "
<construction-expr> ::= SEQ(<construction-term1>[<construction-term2>]...);
<construction-term1> ::= <construction-factor1>|<construction-factor2>;
<construction-term2> ::= <construction-term1>
                        | OPT<construction-factor1>;
<construction-factor1> ::= OBJECT_CLASS_ID_OF(BasicColumn)
                        | OBJECT_CLASS_ID_OF(CompositeFixtureVariable)
                        | OBJECT_CLASS_ID_OF(TableArea);
<construction-factor2> ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                        | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")

DEFINE(CompositeColumnGFS, "
<construction-expr> ::= <construction-term>
                        | <construction-type>
                        | SEQ(<construction-term> <construction-footnote>)
                        | SEQ(<construction-type> <construction-footnote>);
<construction-term> ::= <construction-factor1>
                        | OPT <construction-factor1>
                        | REP <construction-factor1>
                        | OPT REP <construction-factor1>;
<construction-type> ::= SEQ({<construction-term> | <construction-factor2>}...)
                        | CHO({<construction-term>}...);
<construction-factor1> ::= OBJECT_CLASS_ID_OF(BasicFloat)
                        | OBJECT_CLASS_ID_OF(TableArea)
                        | OBJECT_CLASS_ID_OF(CompositeFloat)
                        | OBJECT_CLASS_ID_OF(CompositeFixtureVariable)
                        | OBJECT_CLASS_ID_OF(FooterArea)
                        | <construction-type>;
<construction-footnote> ::= OBJECT_CLASS_ID_OF(FooterArea)
                        | OPT OBJECT_CLASS_ID_OF(FooterArea);
<construction-factor2> ::= OBJECT_CLASS_ID_OF(ArrangedContentVariable)
                        | OBJECT_CLASS_ID_OF(SourcedContentVariable);
")

```

```

DEFINE(CompositeColumnVariableGFS, "$CompositeColumnGFS")
DEFINE(CompositeColumnFixedGFS, "$CompositeColumnGFS")
DEFINE(CompositeFixtureGFS, "
<construction-expr> ::= <construction-factor>
                        | REP CHO(<construction-factor>...);
<construction-factor> ::= OBJECT_CLASS_ID_OF(BasicFloat)
                        | OBJECT_CLASS_ID_OF(FootnoteArea)
                        | OBJECT_CLASS_ID_OF(CompositeArtwork)
                        | OBJECT_CLASS_ID_OF(FormArea);
")
DEFINE(CompositeFixtureFixedGFS, "$CompositeFixtureGFS")
DEFINE(CompositeFixtureVariableGFS, "$CompositeFixtureGFS")
DEFINE(CompositeArtworkGFS, "
<construction-expr> ::= REP OBJECT_CLASS_ID_OF(BasicFixture);
")
DEFINE(TableAreaGFS, "
<construction-expr> ::= <row-area>
                        | SEQ([<table-header>] [<table-label>] <row-area>[<table-label>]);
<table-header> ::= OBJECT_CLASS_ID_OF(TableHeader);
<table-label> ::= OBJECT_CLASS_ID_OF(TableLabel);
<row-area> ::= REP OBJECT_CLASS_ID_OF(RowArea)
            | REP CHO(OBJECT_CLASS_ID_OF(RowArea)...);
")
DEFINE(RowAreaGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(Cell)...)
                        | SEQ(OBJECT_CLASS_ID_OF(Cell)
                            OBJECT_CLASS_ID_OF(SubRowGroup));
")
DEFINE(SubRowGroupGFS, "
<construction-expr> ::= REP OBJECT_CLASS_ID_OF(SubRow);
")
DEFINE(SubRowGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(Cell)...);
")
DEFINE(TableHeaderGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(SourcedContentFixed)...);
")
DEFINE(TableLabelGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)...)
                        | SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)
                            OBJECT_CLASS_ID_OF(CompositeTableLabel));
")
DEFINE(CompositeTableLabelGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(LabelComponent)...);
")
DEFINE(LabelComponentGFS, "
<construction-expr> ::= SEQ(OBJECT_CLASS_ID_OF(TableLabelContent)...);
")
DEFINE(FormAreaGFS, "
<construction-expr> ::= AGG(<factor>...);
<factor> ::= OBJECT_CLASS_ID_OF(ArrangedContentFixed)
            | OBJECT_CLASS_ID_OF(FormEntryArea)
            | OBJECT_CLASS_ID_OF(EntryGroupArea);
")
DEFINE(EntryGroupAreaGFS,"$FormAreaGFS")

```

7.4.2 Factor constraints

7.4.2.1 FACTOR ANY-LAYOUT

```
{
GENERIC:
    REQObject-type           {VIRTUAL},
    REQObject-class-identifier {ANY_VALUE}
SPECIFIC:
    PERM Object-type           {VIRTUAL},
    REQObject-identifier      {ANY_VALUE},
    CASE $DAC OF {
        $FDA: PERM Object-class {VIRTUAL}
        $FPDA: REQ Object-class {VIRTUAL}
    },
    REQSubordinates           {VIRTUAL}
SPECIFIC_AND_GENERIC:
    PERM User-readable-comments {ANY_STRING},
    PERM User-visible-name      {ANY_STRING}
}
```

7.4.2.2 FACTOR ANY-PAGE

```
:ANY-LAYOUT {
GENERIC:
    REQ Object-type           {'page'},
    CASE $DAC OF {
        $PDA-FPDA:
            REQ Generator-for-subordinates {$PageGFS},
            PERM Bindings                 {PMUL {$SPECIFYBINDINGS},
                                           PERM {$INITIALISEPGNUMBER,$USEPGNUMBERS}}
    }
SPECIFIC:
    PERM Object-type           {'page'},
    REQSubordinates            {SUB_ID_OF(BasicHeader),
                               SUB_ID_OF(CompositeHeader),
                               SUB_ID_OF(BasicBody),
                               SUB_ID_OF(FixedCompositeBody),
                               SUB_ID_OF(VariableCompositeBody),
                               SUB_ID_OF(BasicFooter),
                               SUB_ID_OF(CompositeFooter)}
SPECIFIC_AND_GENERIC:
    PERM Dimensions            {$PermissiblePageDimensions},
    PERM Transparency          {ANY_VALUE},
    PERM Colour                 {ANY_VALUE},
    PERM Page-position          {ANY_VALUE}
}
```

7.4.2.3 FACTOR ANY-FRAME-FIXED

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


7.4.2.4 FACTOR ANY-FRAME-VARIABLE

```

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

```

7.4.3 Constituent constraints

7.4.3.1 DocumentLayoutRoot

```

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

```

7.4.3.2 PageSet

```

:ANY-LAYOUT {
GENERIC:
    REQObject-type          {'page-set'},

```

```

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

```

7.4.3.3 Page

```

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

```

7.4.3.4 RectoPage

```

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

```

7.4.3.5 VersoPage

:ANY-PAGE {

GENERIC:

REQApplication-comments {REQ #constraint-name {"4"},
PERM #external-data {ANY_VALUE}},
REQMedium-type {PERM #nominal-page-size {\$NominalPageSizes},
REQ #side-of-sheet {'verso'|'unspecified'}}

SPECIFIC:

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

7.4.3.6 CompositeHeader

:ANY-FRAME-FIXED {

GENERIC:

CASE \$DAC OF {
\$PDA-FPDA:
REQ Generator-for-subordinates {\$HeaderFooterGFS}},
PERM Layout-path {'270-degrees' -- H/F layouts A1,B2 --
|'180-degrees' -- H/F layout B1 --
|'0-degrees' -- H/F layout A2 --},
REQApplication-comments {REQ #constraint-name {"5"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

CASE \$DAC OF {
\$FDA:
PERM Object-class {OBJECT_CLASS_ID_OF(CompositeHeader)}
\$FPDA:
REQ Object-class {OBJECT_CLASS_ID_OF(CompositeHeader)}
},
REQSubordinates {SUB_ID_OF(SourcedContentFixed)+,
SUB_ID_OF(ArrangedContentFixed)+,
SUB_ID_OF(SourcedContentVariable)+,
SUB_ID_OF(ArrangedContentVariable)+},
PERM Imaging-order {SUB_ID_OF(SourcedContentFixed)+,
SUB_ID_OF(ArrangedContentFixed)+},
PERM Application-comments {REQ #constraint-name {"5"},
PERM #external-data {ANY_VALUE}}

7.4.3.7 CompositeFooter

:ANY-FRAME-FIXED {

GENERIC:

CASE \$DAC OF {
\$PDA-FPDA:
REQ Generator-for-subordinates {\$HeaderFooterGFS}},
PERM Layout-path {'270-degrees' -- H/F layouts A1,B2 --
|'180-degrees' -- H/F layout B1 --
|'0-degrees' -- H/F layout A2 --},
REQApplication-comments {REQ #constraint-name {"32"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:

CASE \$DAC OF {
\$FDA:
PERM Object-class {OBJECT_CLASS_ID_OF(CompositeFooter)}
\$FPDA:
REQ Object-class {OBJECT_CLASS_ID_OF(CompositeFooter)}
},

```

REQSubordinates      {SUB_ID_OF(SourcedContentFixed)+,
                      SUB_ID_OF(ArrangedContentFixed)+,
                      SUB_ID_OF(SourcedContentVariable)+,
                      SUB_ID_OF(ArrangedContentVariable)+},
PERM  Imaging-order  {SUB_ID_OF(SourcedContentFixed)+,
                      SUB_ID_OF(ArrangedContentFixed)+},
PERM  Application-comments {REQ #constraint-name {"32"},
                             PERM #external-data {ANY_VALUE}}
}

```

7.4.3.8 FixedCompositeBody

```
:ANY-FRAME-FIXED {
```

```
GENERIC:
```

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ  Generator-for-subordinates  {$FixedCompositeBodyGFS},
    PERM Layout-path      {'270-degrees' -- body layout A --
                          |'0-degrees' -- body layout B --
                          |'180-degrees' -- body layout C --}
  },
REQApplication-comments {REQ #constraint-name {"6"},
                          PERM #external-data {ANY_VALUE}}

```

```
SPECIFIC:
```

```

CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF
                           (FixedCompositeBody)}
  $FPDA:
    REQ  Object-class      {OBJECT_CLASS_ID_OF
                           (FixedCompositeBody)}
  },
REQSubordinates          {SUB_ID_OF(CompositeCommon)+,
                          SUB_ID_OF(BasicFixture)+,
                          SUB_ID_OF(ColumnFixed)+,
                          SUB_ID_OF(CompositeFixtureFixed)+,
                          SUB_ID_OF(VariableCompositeBody)+,
                          SUB_ID_OF(SourcedContentFixed)+,
                          SUB_ID_OF(ArrangedContentFixed)+},
PERM  Imaging-order      {SUB_ID_OF(CompositeCommon)+,
                          SUB_ID_OF(BasicFixture)+,
                          SUB_ID_OF(ColumnFixed)+,
                          SUB_ID_OF(CompositeFixtureFixed)+,
                          SUB_ID_OF(VariableCompositeBody)+,
                          SUB_ID_OF(SourcedContentFixed)+,
                          SUB_ID_OF(ArrangedContentFixed)+},
PERM  Application-comments {REQ #constraint-name {"6"},
                             PERM #external-data {ANY_VALUE}}
}

```

7.4.3.9 VariableCompositeBody

```
:ANY-FRAME-FIXED {
```

```
GENERIC:
```

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ  Generator-for-subordinates  {$VariableCompositeBodyGFS},
    PERM Layout-path      {'270-degrees' -- body layout A --
                          |'0-degrees' -- body layout B --
                          |'180-degrees' -- body layout C --}
  },
REQApplication-comments {REQ #constraint-name {"7"},
                          PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF
                        (VariableCompositeBody)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF
                     (VariableCompositeBody)}
},
REQSubordinates {SUB_ID_OF(SnakingColumns)+,
                  SUB_ID_OF(SynchronizedColumns)+,
                  SUB_ID_OF(BasicFloat)+,
                  SUB_ID_OF(FootnoteArea)+,
                  SUB_ID_OF(CompositeFloat)+,
                  SUB_ID_OF(CompositeFixtureVariable)+,
                  SUB_ID_OF(TableArea)+,
                  SUB_ID_OF(ArrangedContentVariable)+,
                  SUB_ID_OF(SourcedContentVariable)+},
PERM Application-comments {REQ #constraint-name {"7"},
                           PERM #external-data {ANY_VALUE}}
}

```

7.4.3.10 ColumnFixed

:ANY-FRAME-VARIABLE {

GENERIC:

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Position {REQ #fixed-position
                  {REQ #horizontal-position {ANY_VALUE},
                   REQ #vertical-position {ANY_VALUE}}},
    CASE SUPERIOR ({VariableCompositeBody
                  | FixedCompositeBody} (Layout-path)) OF {
      {'270-degrees'}:-- body layout A --
        REQ Dimensions {REQ #horizontal-dimension
                        {REQ #fixed-dimension {ANY_VALUE}
                         |REQ #maximum-size {'applies'}},
                        REQ #vertical-dimension
                        {REQ #rule-b {ANY_VALUE}
                         |REQ #maximum-size {'applies'}}},
        PERM Layout-path {'270-degrees'}
      {'0-degrees'}:-- body layout B --
        REQ Dimensions {REQ #horizontal-dimension
                        {REQ #rule-b {ANY_VALUE}
                         |REQ #maximum-size {'applies'}},
                        REQ #vertical-dimension
                        {REQ #fixed-dimension {ANY_VALUE}
                         |REQ #maximum-size {'applies'}}},
        REQ Layout-path {'0-degrees'}
      {'180-degrees'}:-- body layout C --
        REQ Dimensions {REQ #horizontal-dimension
                        {REQ #maximum-size {'applies'}},
                        REQ #vertical-dimension
                        {REQ #fixed-dimension {ANY_VALUE}
                         |REQ #maximum-size {'applies'}}},
        REQ Layout-path {'180-degrees'}
    } },
    REQApplication-comments {REQ #constraint-name {"8"},
                            PERM #external-data {ANY_VALUE}}
}

```

SPECIFIC:

```

CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF(ColumnFixed)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(ColumnFixed)}
},

```

```

REQ Subordinates {SUB_ID_OF(SpecificBlock)+},
PERM Imaging-order {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"8"},
PERM #external-data {ANY_VALUE}}

```

SPECIFIC_AND_GENERIC:

```

PERM Permitted-categories {ANY_STRING...}

```

}

7.4.3.11 ColumnVariable

:ANY-FRAME-VARIABLE {
GENERIC:

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Position {REQ #variable-position {
      PERM #offset {ANY_VALUE},
      PERM #separation {ANY_VALUE},
      PERM #alignment {ANY_VALUE},
      PERM #fill-order {'normal-order'}},
    CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {

```

```

{'270-degrees': -- body layout A --

```

```

    REQ Dimensions {REQ #horizontal-dimension
      {REQ #fixed-dimension {ANY_VALUE}},
      REQ #vertical-dimension
      {REQ #rule-b {ANY_VALUE}
      |REQ #maximum-size {'applies'}},
    PERM Layout-path {'270-degrees'}

```

```

{'0-degrees': -- body layout B --

```

```

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

```

```

{'180-degrees': -- body layout C --

```

```

    REQ Dimensions {REQ #horizontal-dimension
      {REQ #rule-b {ANY_VALUE}
      |REQ #maximum-size {'applies'}},
      REQ #vertical-dimension
      {REQ #fixed-dimension {ANY_VALUE}}},
    REQ Layout-path {'180-degrees'}

```

```

} },

```

```

REQ Application-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF(ColumnVariable)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(ColumnVariable)}
},

```

```

REQSubordinates {SUB_ID_OF(SpecificBlock)+},
PERM Imaging-order {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"9"},
PERM #external-data {ANY_VALUE}}

```

SPECIFIC_AND_GENERIC:

```

PERM Permitted-categories {ANY_STRING...}

```

}

7.4.3.12 SnakingColumns

:ANY-FRAME-VARIABLE {
GENERIC:

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates    {$SnakingColumnsGFS},
    REQ Position                      {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},

    PERM Balance                      {ANY_VALUE},

  CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
    {'270-degrees'}: -- body layout A --
      REQ Dimensions                  {REQ #horizontal-dimension
      {REQ #fixed-dimension {ANY_VALUE}
      |REQ #maximum-size {'applies'}},
      REQ #vertical-dimension
      {REQ #rule-b {ANY_VALUE}}},
      REQ Layout-path                 {'0-degrees'|'180-degrees'}

    {'0-degrees'}: -- body layout B --
      REQ Dimensions                  {REQ #horizontal-dimension
      {REQ #rule-b {ANY_VALUE}},
      REQ #vertical-dimension
      {REQ #fixed-dimension {ANY_VALUE}
      |REQ #maximum-size {'applies'}}},
      PERM Layout-path                 {'90-degrees'|'270-degrees'}

    {'180-degrees'}: -- body layout C --
      REQ Dimensions                  {REQ #horizontal-dimension
      {REQ #rule-b {ANY_VALUE}},
      REQ #vertical-dimension
      {REQ #fixed-dimension {ANY_VALUE}
      |REQ #maximum-size {'applies'}}},
      PERM Layout-path                 {'270-degrees'}

  } },
  REQApplication-comments             {REQ #constraint-name {'10"},
  PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

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

```

7.4.3.13 SynchronizedColumns

:ANY-FRAME-VARIABLE {
GENERIC:

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates    {$SynchronizedColumnsGFS},
    REQ Position                      {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},

```

```

CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
  {'270-degrees': -- body layout A --
    REQ Dimensions          {REQ #horizontal-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}},
                             REQ #vertical-dimension
                             {REQ #rule-b {ANY_VALUE}}},
    PERM Layout-path        {'270-degrees'}

  {'0-degrees': -- body layout B --
    REQ Dimensions          {REQ #horizontal-dimension
                             {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
    REQ Layout-path         {'0-degrees'}

  {'180-degrees': -- body layout C --
    REQ Dimensions          {REQ #horizontal-dimension
                             {REQ #rule-b {ANY_VALUE}},
                             REQ #vertical-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}}},
    REQ Layout-path         {'180-degrees'}
  } },
REQApplication-comments    {REQ #constraint-name {"11"},
                             PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
  $FDA:
    PERM Object-class OBJECT_CLASS_ID_OF(SynchronizedColumns)
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(SynchronizedColumns)}
  },
REQSubordinates           {SUB_ID_OF(ColumnFixed)+,
                           SUB_ID_OF(CompositeColumnFixed)+,
                           SUB_ID_OF(ArrangedContentFixed)+,
                           SUB_ID_OF(SourcedContentFixed)+},
PERM Application-comments {REQ #constraint-name {"11"},
                             PERM #external-data {ANY_VALUE}}
}

```

7.4.3.14 BasicFloat

:ANY-FRAME-VARIABLE {
GENERIC:

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Position           {REQ #variable-position {
                             PERM #offset {ANY_VALUE},
                             PERM #separation {ANY_VALUE},
                             PERM #alignment {ANY_VALUE},
                             PERM #fill-order {'normal-order'}}},

CASE SUPERIOR ({VariableCompositeBody
  | CompositeColumnVariable | CompositeColumnFixed
  | CompositeFixtureVariable | CompositeFixtureFixed} (Layout-path)) OF {
  {'270-degrees': -- body layout A --
    REQ Dimensions          {REQ #horizontal-dimension
                             {REQ #fixed-dimension {ANY_VALUE}
                              |REQ #maximum-size {'applies'}},
                             REQ #vertical-dimension
                             {REQ #rule-b {ANY_VALUE}}},
    PERM Layout-path        {'270-degrees'}

```



```

{'0-degrees': -- body layout B --
    REQ Dimensions          {REQ #horizontal-dimension
                            {REQ #rule-b {ANY_VALUE}},
                            REQ #vertical-dimension
                            {REQ #fixed-dimension {ANY_VALUE}
                            |REQ #maximum-size {'applies'}}},
    REQ Layout-path        {'0-degrees'}
{'180-degrees': -- body layout C --
    REQ Dimensions          {REQ #horizontal-dimension
                            {REQ #rule-b {ANY_VALUE}},
                            REQ #vertical-dimension
                            {REQ #fixed-dimension {ANY_VALUE}
                            |REQ #maximum-size {'applies'}}},
    REQ Layout-path        {'180-degrees'}
} },
REQApplication-comments    {REQ #constraint-name {"12"},
                            PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
    $FDA:
        PERM Object-class    {OBJECT_CLASS_ID_OF(BasicFloat)}
    $FPDA:
        REQ Object-class     {OBJECT_CLASS_ID_OF(BasicFloat)}
    },
REQSubordinates            {SUB_ID_OF(SpecificBlock)+},
PERM Imaging-order        {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments  {REQ #constraint-name {"12"},
                            PERM #external-data {ANY_VALUE}}

```

SPECIFIC_AND_GENERIC:

```

PERM Permitted-categories  {ANY_STRING...}
}

```

7.4.3.15 CompositeFloat

:ANY-FRAME-VARIABLE {

GENERIC:

```

CASE $DAC OF {
    $PDA-FPDA:
        REQ Generator-for-subordinates    {$CompositeFloatGFS},
        REQ Position                      {REQ #variable-position {
                                            PERM #offset {ANY_VALUE},
                                            PERM #separation {ANY_VALUE},
                                            PERM #alignment {ANY_VALUE},
                                            PERM #fill-order {'normal-order'}}},

```

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

```

{'270-degrees': -- body layout A --
    REQ Dimensions          {REQ #horizontal-dimension
                            {REQ #fixed-dimension {ANY_VALUE}
                            |REQ #maximum-size {'applies'}},
                            REQ #vertical-dimension
                            {REQ #rule-a {ANY_VALUE}}},
    PERM Layout-path        {'0-degrees'|'180-degrees'}
{'0-degrees': -- body layout B --
    REQ Dimensions          {REQ #horizontal-dimension
                            {REQ #rule-a {ANY_VALUE}},
                            REQ #vertical-dimension
                            {REQ #fixed-dimension {ANY_VALUE}
                            |REQ #maximum-size {'applies'}}},
    REQ Layout-path        {'90-degrees'|'270-degrees'}

```

```

        {'180-degrees': -- body layout C --
          REQ Dimensions          {REQ #horizontal-dimension
                                  {REQ #rule-a {ANY_VALUE}},
                                  REQ #vertical-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}}},
          REQ Layout-path        {'90-degrees'|'270-degrees'}
        } },
    REQApplication-comments      {REQ #constraint-name {"13"},
                                  PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class    {OBJECT_CLASS_ID_OF(CompositeFloat)}
        $FPDA:
            REQ Object-class     {OBJECT_CLASS_ID_OF(CompositeFloat)}
    },
    REQSubordinates             {SUB_ID_OF(CompositeFixtureVariable)+,
                                  SUB_ID_OF(TableArea)+,
                                  SUB_ID_OF(BasicColumn)+,
                                  SUB_ID_OF(ArrangedContentVariable)+,
                                  SUB_ID_OF(SourcedContentVariable)+},
    PERM Application-comments   {REQ #constraint-name {"13"},
                                  PERM #external-data {ANY_VALUE}}
}

```

7.4.3.16 BasicColumn

:ANY-FRAME-VARIABLE {
GENERIC:

```

    CASE $DAC OF {
        $PDA-FPDA:
            REQ Position          {REQ #variable-position {
                                    PERM #offset {ANY_VALUE},
                                    PERM #separation {ANY_VALUE},
                                    PERM #alignment {ANY_VALUE},
                                    PERM #fill-order {'normal-order'}}},
    }

```

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

```

    {'270-degrees': -- body layout A --
        REQ Dimensions          {REQ #horizontal-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}},
                                  REQ #vertical-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}}},
        PERM Layout-path        {'270-degrees'}
    }
    {'0-degrees': -- body layout B --
        REQ Dimensions          {REQ #horizontal-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size{'applies'}},
                                  REQ #vertical-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}}},
        REQ Layout-path          {'0-degrees'}
    }
    {'180-degrees': -- body layout C --
        REQ Dimensions          {REQ #horizontal-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}
                                   |REQ #maximum-size{'applies'}},
                                  REQ #vertical-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}}},
        REQ Layout-path          {'180-degrees'}
    } },

```

```

REQ Application-comments {REQ #constraint-name {"14"},
                          PERM #external-data {ANY_VALUE}}
SPECIFIC:
CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF(BasicColumn)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(BasicColumn)}
  },
REQSubordinates {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"14"},
                           PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
PERM Permitted-categories {ANY_STRING...}
}

```

7.4.3.17 FootnoteArea

```

:ANY-FRAME-VARIABLE {
GENERIC:

```

```

CASE $DAC OF {
  $PDA-FPDA:
    RQ Position {REQ #variable-position {
                  PERM #offset {ANY_VALUE},
                  PERM #separation {ANY_VALUE},
                  PERM #alignment {ANY_VALUE},
                  PERM #fill-order {'reverse-order'}}},

CASE SUPERIOR ({VariableCompositeBody
  | CompositeColumnVariable | CompositeColumnFixed
  | CompositeFixtureVariable | CompositeFixtureFixed} (Layout-path)) OF {
{'270-degrees': -- body layout A --
  REQ Dimensions {REQ #horizontal-dimension
                  {REQ #fixed-dimension {ANY_VALUE}
                  |REQ #maximum-size {'applies'}},
                  REQ #vertical-dimension
                  {REQ #rule-b {ANY_VALUE}}},
  PERM Layout-path {'270-degrees'}

{'0-degrees': -- body layout B --
  REQ Dimensions {REQ #horizontal-dimension
                  {REQ #rule-b {ANY_VALUE}}},
                  REQ #vertical-dimension
                  {REQ #fixed-dimension {ANY_VALUE}
                  |REQ #maximum-size {'applies'}},
  REQ Layout-path {'0-degrees'}

{'180-degrees': -- body layout C --
  REQ Dimensions {REQ #horizontal-dimension
                  {REQ #rule-b {ANY_VALUE}}},
                  REQ #vertical-dimension
                  {REQ #fixed-dimension {ANY_VALUE}
                  |REQ #maximum-size {'applies'}},
  REQ Layout-path {'180-degrees'}
} },
REQPermitted-categories {$FOOTNOTECATEGORY},
-- For example,
  For CompositeBody "Footnote-1"
  For SnakingColumns "Footnote-2"
  For SynchronizedColumns "Footnote-3", "Footnote-4",
                          "Footnote-5"
  For CompositeFixture "Footnote-6"
--
REQ Application-comments {REQ #constraint-name {"15"},
                          PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```
    CASE $DAC OF {
      $FDA:
        PERM Object-class          {OBJECT_CLASS_ID_OF(FootnoteArea)}
      $FPDA:
        REQ Object-class           {OBJECT_CLASS_ID_OF(FootnoteArea)}
    },
    REQSubordinates                {SUB_ID_OF(SpecificBlock)+},
    PERM Permitted-categories       {$FOOTNOTECATEGORY},
    PERM Application-comments       {REQ #constraint-name {"15"},
                                     PERM #external-data {ANY_VALUE}}
  }
```

7.4.3.18 ArrangedContentFixed

:ANY-FRAME-FIXED {

GENERIC:

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

SPECIFIC:

```
    CASE $DAC OF {
      $FDA:
        PERM Object-class          {OBJECT_CLASS_ID_OF
                                     (ArrangedContentFixed)}
      $FPDA:
        REQ Object-class           {OBJECT_CLASS_ID_OF
                                     (ArrangedContentFixed)}
    },
    REQSubordinates                {SUB_ID_OF(GenericBlock)+},
    PERM Imaging-order              {SUB_ID_OF(GenericBlock)+},
    PERM Application-comments       {REQ #constraint-name {"16"},
                                     PERM #external-data {ANY_VALUE}}
  }
```

7.4.3.19 ArrangedContentVariable

:ANY-FRAME-VARIABLE {

GENERIC:

```
    CASE $DAC OF {
      $PDA-FPDA:
        REQ Generator-for-subordinates {<construction-expr>::=
SEQ(OBJECT_CLASS_ID_OF(GenericBlock)...)},
        REQ Position                  {REQ #variable-position {
PERM #offset {ANY_VALUE},
PERM #separation {ANY_VALUE},
PERM #alignment {ANY_VALUE},
PERM #fill-order {'normal-order'}}},
        REQ Dimensions                {REQ #horizontal-dimension
{REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
{REQ #fixed-dimension {ANY_VALUE}}}
    },
    REQApplication-comments          {REQ #constraint-name {"17"},
                                     PERM #external-data {ANY_VALUE}}
  }
```

SPECIFIC:

```
    CASE $DAC OF {
      $FDA:
        PERM Object-class          {OBJECT_CLASS_ID_OF
                                     (ArrangedContentVariable)}
      $FPDA:
        REQ Object-class           {OBJECT_CLASS_ID_OF
                                     (ArrangedContentVariable)}
    },
  }
```

```

REQSubordinates      {SUB_ID_OF(GenericBlock)+},
PERM Imaging-order   {SUB_ID_OF(GenericBlock)+},
PERM Application-comments {REQ #constraint-name {"17"},
                        PERM #external-data {ANY_VALUE}}

```

}

7.4.3.20 SourcedContentFixed

```

:ANY-FRAME-VARIABLE {
GENERIC:

```

```

CASE $DAC OF {
  $PDA-FPDA:
    REQ Logical-source {OBJECT_CLASS_ID_OF(CommonContent)},
    REQ Position       {REQ #fixed-position
                        {REQ #horizontal-position {ANY_VALUE},
                         REQ #vertical-position {ANY_VALUE}}},
    REQ Dimensions     {REQ #horizontal-dimension
                        {REQ #fixed-dimension {ANY_VALUE}},
                         REQ #vertical-dimension
                        {REQ #fixed-dimension {ANY_VALUE}}},

```

```

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

```

```

{'270-degrees': -- H/F layout A1 or B2 when the immediate superior is CompositeHeader, CompositeFooter
or TableHeader, or --
                -- body layout A when the immediate superior is FixedCompositeBody,
CompositeCommon or
SynchronizedColumns --

```

```

    PERM Layout-path {'270-degrees' -- H/F layout A1 or body layout A --
|'180-degrees'} -- H/F layout B2 --

```

```

{'180-degrees': -- H/F layout B1 when the immediate superior is CompositeHeader or CompositeFooter, or
-- body layout C when the immediate superior is FixedCompositeBody, CompositeCommon
or SynchronizedColumns --

```

```

    REQ Layout-path {'180-degrees'} -- H/F layout B1 or body layout C --

```

```

{'0-degrees': -- H/F layout A2 when the immediate superior is CompositeHeader or CompositeFooter, o
-- body layout B when the immediate superior is FixedCompositeBody, CompositeCommon or
SynchronizedColumns --

```

```

    PERM Layout-path {'270-degrees' -- H/F layout A2 --
|'0-degrees'} -- body layout B --

```

```

} },

```

```

REQApplication-comments {REQ #constraint-name {"18"},
                        PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF
                      (SourcedContentFixed)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF
                    (SourcedContentFixed)}

```

```

},

```

```

REQSubordinates      {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"18"},
                        PERM #external-data {ANY_VALUE}}

```

}

7.4.3.21 SourcedContentVariable

```

:ANY-FRAME-VARIABLE {
GENERIC:

```

```

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

```

CASE SUPERIOR ({CompositeHeader | CompositeFooter | VariableCompositeBody
 | CompositeColumnVariable | CompositeColumnFixed | CompositeCommon
 | SnakingColumns | CompositeFloat} (Layout-path)) OF {

{'270-degrees': -- H/F layout A1 or B2 when the immediate superior is CompositeHeader or
 CompositeFooter, or --
 -- body layout A when the immediate superior is VariableCompositeBody,
 CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
 -- body layout B or C when the immediate superior is SnakingColumns or

CompositeFloat --

REQ Dimensions {REQ #horizontal-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}},
 REQ #vertical-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}}},
 PERM Layout-path {'270-degrees' -- H/F layout A1 or body layout A --
 |'180-degrees' -- H/F layout B2 or body layout C --
 |'0-degrees' -- body layout B --

{'180-degrees': -- H/F layout B1 when the immediate superior is CompositeHeader or CompositeFooter, or --
 -- body layout C when the immediate superior is VariableCompositeBody,
 CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
 -- body layout A when the immediate superior is SnakingColumns or

CompositeFloat --

REQ Dimensions {REQ #horizontal-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}},
 REQ #vertical-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}}},
 REQ Layout-path {'180-degrees' -- H/F layout B1 or body layout C --
 |'270-degrees' -- body layout A --

{' 0-degrees': -- H/F layout A2 when the immediate superior is CompositeHeader or CompositeFooter, or --
 -- body layout B when the immediate superior is VariableCompositeBody,
 CompositeColumnVariable, CompositeColumnFixed or CompositeCommon, or --
 -- body layout A when the immediate superior is SnakingColumns or
 CompositeFloat --

REQ Dimensions {REQ #horizontal-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}},
 REQ #vertical-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}}},
 PERM Layout-path {'270-degrees' -- H/F layout A2 or body layout A --
 |'0-degrees' -- body layout B --

{'90-degrees': -- body layout B when the immediate superior is SnakingColumns, or --
 -- body layout B or C when the immediate superior is CompositeFloat --

REQ Dimensions {REQ #horizontal-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}
 |REQ #maximum-size {'applies'}},
 REQ #vertical-dimension
 {REQ #fixed-dimension {ANY_VALUE}
 |REQ #rule-b {ANY_VALUE}},
 PERM Layout-path {'0-degrees' -- body layout B --
 |'180-degrees' -- body layout C --

} },

REQ Application-comments {REQ #constraint-name {"19"},
 PERM #external-data {ANY_VALUE}}

SPECIFIC:

```
    CASE $DAC OF {
      $FDA:
        PERM Object-class          {OBJECT_CLASS_ID_OF
                                   (SourcedContentVariable)}

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

7.4.3.22 CompositeFixtureVariable

:ANY-FRAME-VARIABLE {

GENERIC:

```
    CASE $DAC OF {
      $PDA-FPDA:
        REQ Generator-for-subordinates  {$CompositeFixtureVariableGFS},
        REQ Position                    {REQ #variable-position {
                                         PERM #offset {ANY_VALUE},
                                         PERM #separation {ANY_VALUE},
                                         PERM #alignment {ANY_VALUE},
                                         PERM #fill-order {'normal-order'}}},

      CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
        {'270-degrees'}: -- body layout A --
          REQ Dimensions                {REQ #horizontal-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #maximum-size {'applies'}
                                          |REQ #rule-b {ANY_VALUE}},
                                         REQ #vertical-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #rule-b {ANY_VALUE}}},
          PERM Layout-path              {'270-degrees'|'180-degrees'
                                         |'0-degrees'}

        {'0-degrees'}: -- body layout B --
          REQ Dimensions                {REQ #horizontal-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #rule-b {ANY_VALUE}},
                                         REQ #vertical-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #maximum-size {'applies'}
                                          |REQ #rule-b {ANY_VALUE}}},
          REQ Layout-path               {'0-degrees'|'90-degrees'
                                         |'270-degrees'}

        {'180-degrees'}: -- body layout C --
          REQ Dimensions                {REQ #horizontal-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #rule-b {ANY_VALUE}},
                                         REQ #vertical-dimension
                                         {REQ #fixed-dimension {ANY_VALUE}
                                          |REQ #maximum-size {'applies'}
                                          |REQ #rule-b {ANY_VALUE}}},
          REQ Layout-path               {'180-degrees'|'270-degrees'}
      } },
    REQ Application-comments            {REQ #constraint-name {"20"},
                                         PERM #external-data {ANY_VALUE}}
```

SPECIFIC:

```
CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF
                           (CompositeFixtureVariable)}
    $FPDA:
    REQ Object-class       {OBJECT_CLASS_ID_OF
                           (CompositeFixtureVariable)}
  },
REQ Subordinates          {SUB_ID_OF(BasicFloat)+,
                           SUB_ID_OF(FootnoteArea)+,
                           SUB_ID_OF(CompositeArtwork)+,
                           SUB_ID_OF(FormArea)+},
  PERM Application-comments {REQ #constraint-name {"20"},
                             PERM #external-data {ANY_VALUE}}
}
```

7.4.3.23 CompositeFixtureFixed

:ANY-FRAME-VARIABLE {

GENERIC:

```
CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates {$CompositeFixtureFixedGFS},
    REQ Position                   {REQ #fixed-position
                                   {REQ #horizontal-position {ANY_VALUE},
                                   REQ #vertical-position {ANY_VALUE}}},
    CASE SUPERIOR (FixedCompositeBody(Layout-path)) OF {
      {'270-degrees': -- body layout A --
        REQ Dimensions              {REQ #horizontal-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}},
                                   REQ #vertical-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}}},
        PERM Layout-path            {'270-degrees'|'180-degrees'
                                   |'0-degrees'}
      {'0-degrees': -- body layout B --
        REQ Dimensions              {REQ #horizontal-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}},
                                   REQ #vertical-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}}},
        REQ Layout-path             {'0-degrees'|'90-degrees'
                                   |'270-degrees'}
      {'180-degrees': -- body layout C --
        REQ Dimensions              {REQ #horizontal-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #rule-b {ANY_VALUE}},
                                   REQ #vertical-dimension
                                   {REQ #fixed-dimension {ANY_VALUE}
                                   |REQ #maximum-size {'applies'}}},
        REQ Layout-path             {'180-degrees'|'270-degrees'}
      } },
    REQ Application-comments        {REQ #constraint-name {"21"},
                                     PERM #external-data {ANY_VALUE}}
}
```

SPECIFIC:

```
CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF
                           (CompositeFixtureFixed)}
    $FPDA:
    REQ Object-class       {OBJECT_CLASS_ID_OF
                           (CompositeFixtureFixed)}
  },
```



```

REQ Subordinates      {SUB_ID_OF(BasicFloat)+,
                      SUB_ID_OF(FootnoteArea)+,
                      SUB_ID_OF(CompositeArtwork)+,
                      SUB_ID_OF(FormArea)+},
PERM Application-comments {REQ #constraint-name {"21"},
                          PERM #external-data {ANY_VALUE}}
}

```

7.4.3.24 BasicFixture

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Position      {REQ #fixed-position
                        {REQ #horizontal-position {ANY_VALUE},
                         REQ #vertical-position {ANY_VALUE}}},

      -- Note that values of position may usually be "0" for overlapping figure. --

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

      CASE SUPERIOR ({FixedCompositeBody | CompositeArtwork} (Layout-path)) OF {
        {'270-degrees': -- body layout A --
          PERM Layout-path    {'270-degrees'|'180-degrees'}

        {'0-degrees': -- body layout B --
          REQ Layout-path     {'0-degrees'|'270-degrees'}

        {'180-degrees': -- body layout C --
          PERM Layout-path    {'180-degrees'|'270-degrees'}
      } },
      REQ Application-comments {REQ #constraint-name {"22"},
                              PERM #external-data {ANY_VALUE}}
    },
SPECIFIC:
  CASE $DAC OF {
    $FDA:
      PERM Object-class      {OBJECT_CLASS_ID_OF(BasicFixture)}
    $FPDA:
      REQ Object-class       {OBJECT_CLASS_ID_OF(BasicFixture)}
    },
    REQ Subordinates        {SUB_ID_OF(SpecificBlock)+,
    PERM Application-comments {REQ #constraint-name {"22"},
                              PERM #external-data {ANY_VALUE}}
  },
SPECIFIC_AND_GENERIC:
  PERM Permitted-categories {ANY_STRING...}
}

```

7.4.3.25 CompositeColumnFixed

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates {$CompositeColumnFixedGFS},
      REQ Position      {REQ #fixed-position
                        {REQ #horizontal-position {ANY_VALUE},
                         REQ #vertical-position {ANY_VALUE}}},

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

```

```

    {'270-degrees': -- body layout A --
      REQ Dimensions      {REQ #horizontal-dimension
                          {REQ #fixed-dimension {ANY_VALUE}
                          |REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                          {REQ #rule-b {ANY_VALUE}
                          |REQ #maximum-size {'applies'}}},
      PERM Layout-path    {'270-degrees'}

    {'0-degrees': -- body layout B --
      REQ Dimensions      {REQ #horizontal-dimension
                          {REQ #rule-b {ANY_VALUE}
                          |REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}
                          |REQ #maximum-size {'applies'}}},
      REQ Layout-path     {'0-degrees'}

    {'180-degrees': -- body layout C --
      REQ Dimensions      {REQ #horizontal-dimension
                          {REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}
                          |REQ #maximum-size {'applies'}}},
      REQ Layout-path     {'180-degrees'}

  } },
  REQ Application-comments {REQ #constraint-name {"23"},
                            PERM #external-data {ANY_VALUE}}
SPECIFIC:
  CASE $DAC OF {
    $FDA:
      PERM Object-class    {OBJECT_CLASS_ID_OF
                          (CompositeColumnFixed)}

    $FPDA:
      REQ Object-class     {OBJECT_CLASS_ID_OF
                          (CompositeColumnFixed)}

  },
  REQ Subordinates        {SUB_ID_OF(BasicFloat)+,
                          SUB_ID_OF(CompositeFloat)+,
                          SUB_ID_OF(FootnoteArea)+,
                          SUB_ID_OF(CompositeFixtureVariable)+,
                          SUB_ID_OF(TableArea)+,
                          SUB_ID_OF(ArrangedContentVariable)+,
                          SUB_ID_OF(SourcedContentVariable)+},
  PERM Application-comments {REQ #constraint-name {"23"},
                            PERM #external-data {ANY_VALUE}}
}

```

7.4.3.26 CompositeColumnVariable

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates {$CompositeColumnVariableGFS},
      REQ Position                  {REQ #variable-position {
                                    PERM #offset {ANY_VALUE},
                                    PERM #separation {ANY_VALUE},
                                    PERM #alignment {ANY_VALUE},
                                    PERM #fill-order {'normal-order'}}},

    CASE SUPERIOR (VariableCompositeBody(Layout-path)) OF {
      {'270-degrees': -- body layout A --
        REQ Dimensions      {REQ #horizontal-dimension
                            {REQ #fixed-dimension {ANY_VALUE}},
                            REQ #vertical-dimension
                            {REQ #rule-b {ANY_VALUE}
                            |REQ #maximum-size {'applies'}}},
        PERM Layout-path    {'270-degrees'}

```

```

    {'0-degrees': -- body layout B --
      REQ Dimensions      {REQ #horizontal-dimension
                          {REQ #rule-b {ANY_VALUE}
                          |REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}}},
      REQ Layout-path     {'0-degrees'}
    {'180-degrees': -- body layout C --
      REQ Dimensions      {REQ #horizontal-dimension
                          {REQ #rule-b {ANY_VALUE}
                          |REQ #maximum-size {'applies'}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}}},
      REQ Layout-path     {'180-degrees'}
  } },
  REQ Application-comments {REQ #constraint-name {"24"},
                           PERM #external-data {ANY_VALUE}}
SPECIFIC:
CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF
                           (CompositeColumnVariable)}
  $FPDA:
    REQ Object-class       {OBJECT_CLASS_ID_OF
                           (CompositeColumnVariable)}
},
REQ Subordinates          {SUB_ID_OF(BasicFloat)+,
                           SUB_ID_OF(CompositeFloat)+,
                           SUB_ID_OF(FootnoteArea)+,
                           SUB_ID_OF(CompositeFixtureVariable)+,
                           SUB_ID_OF(TableArea)+,
                           SUB_ID_OF(ArrangedContentVariable)+,
                           SUB_ID_OF(SourcedContentVariable)+},
  PERM Application-comments {REQ #constraint-name {"24"},
                              PERM #external-data {ANY_VALUE}}
}

```

7.4.3.27 CompositeCommon

```

:ANY-FRAME-FIXED {
GENERIC:
CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates {$CompositeCommonGFS},
    CASE SUPERIOR (FixedCompositeBody(Layout-path)) OF {
      {'270-degrees': -- body layout A --
        PERM Layout-path      {'270-degrees'}
      {'0-degrees': -- body layout B --
        PERM Layout-path      {'0-degrees'}
      {'180-degrees': -- body layout C --
        PERM Layout-path      {'180-degrees'}
    } },
  REQ Application-comments   {REQ #constraint-name {"25"},
                              PERM #external-data {ANY_VALUE}}
}

```

```

SPECIFIC:
CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF(CompositeCommon)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(CompositeCommon)}
},
REQ Subordinates      {SUB_ID_OF(SourcedContentFixed)+,
                       SUB_ID_OF(ArrangedContentFixed)+,
                       SUB_ID_OF(SourcedContentVariable)+,
                       SUB_ID_OF(ArrangedContentVariable)+},

```

```

    PERM Imaging-order      {SUB_ID_OF(SourcedContentFixed)+,
                             SUB_ID_OF(ArrangedContentFixed)+},
    PERM Application-comments {REQ #constraint-name {"25"},
                             PERM #external-data {ANY_VALUE}}
}

```

7.4.3.28 CompositeArtwork

```

:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ Generator-for-subordinates    {$CompositeArtworkGFS},
            REQ Position                      {REQ #variable-position {
                PERM #offset {ANY_VALUE},
                PERM #separation {ANY_VALUE},
                PERM #alignment {ANY_VALUE},
                PERM #fill-order {'normal-order'}}},

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

            CASE SUPERIOR ({CompositeFixtureVariable | CompositeFixtureFixed}
                (Layout-path)) OF {
                {'270-degrees':
                    PERM Layout-path    {'270-degrees'}
                {'0-degrees':
                    REQ Layout-path     {'0-degrees'}
                {'180-degrees':
                    REQ Layout-path     {'180-degrees'}
                }},
            REQ Application-comments        {REQ #constraint-name {"26"},
                PERM #external-data {ANY_VALUE}}
    }
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class {OBJECT_CLASS_ID_OF(CompositeArtwork)}
        $FPDA:
            REQ Object-class {OBJECT_CLASS_ID_OF(CompositeArtwork)}
    },
    REQ Subordinates          {SUB_ID_OF(BasicFixture)+},
    PERM Imaging-order        {SUB_ID_OF(BasicFixture)+},
    PERM Application-comments {REQ #constraint-name {"26"},
        PERM #external-data {ANY_VALUE}}
}

```

7.4.3.29 BasicHeader

```

:ANY-FRAME-FIXED {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ Logical-source    {OBJECT_CLASS_ID_OF(CommonContent)}
    },
    PERM Layout-path            {'270-degrees' -- H/F layout A1 --
        {'180-degrees' -- H/F layout B1 --},
    REQ Application-comments    {REQ #constraint-name {"27"},
        PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class    {OBJECT_CLASS_ID_OF(BasicHeader)}
        $FPDA:
            REQ Object-class     {OBJECT_CLASS_ID_OF(BasicHeader)}
    },
}

```

```

REQ Subordinates {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"27"},
PERM #external-data {ANY_VALUE}}
}

```

7.4.3.30 BasicFooter

```

:ANY-FRAME-FIXED {
GENERIC:
CASE $DAC OF {
$PDA-FPDA:
REQ Logical-source {OBJECT_CLASS_ID_OF(CommonContent)}
},
PERM Layout-path {'270-degrees' -- H/F layout A1 --
|'180-degrees' -- H/F layout B1 --},
REQ Application-comments {REQ #constraint-name {"33"},
PERM #external-data {ANY_VALUE}}
SPECIFIC:
CASE $DAC OF {
$FDA:
PERM Object-class {OBJECT_CLASS_ID_OF(BasicFooter)}
$FPDA:
REQ Object-class {OBJECT_CLASS_ID_OF(BasicFooter)}
},
REQ Subordinates {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"33"},
PERM #external-data {ANY_VALUE}}
}

```

7.4.3.31 BasicBody

```

:ANY-FRAME-FIXED {
GENERIC:
PERM Layout-path {'270-degrees' -- body layout A --
|'0-degrees' -- body layout B --
|'180-degrees' -- body layout C --},
REQ Application-comments {REQ #constraint-name {"28"},
PERM #external-data {ANY_VALUE}}
SPECIFIC:
CASE $DAC OF {
$FDA:
PERM Object-class {OBJECT_CLASS_ID_OF(BasicBody)}
$FPDA:
REQ Object-class {OBJECT_CLASS_ID_OF(BasicBody)}
},
REQ Subordinates {SUB_ID_OF(SpecificBlock)+},
PERM Application-comments {REQ #constraint-name {"28"},
PERM #external-data {ANY_VALUE}}
}

```

7.4.3.32 GenericBlock

```

{
GENERIC:
REQ Object-type {'block'},
REQ Object-class-identifier {ANY_VALUE},
REQ Content-architecture-class {$FC | $FPC | $FPR | $FPG },
PERM Content-generator {$GENERICBLOCKREF},
PERM Content-portions {CONTENT_ID_OF(Character-content-portion)+
| CONTENT_ID_OF(Raster-graphics-content-portion)
| CONTENT_ID_OF(Geometric-graphics-content-portion) },
PERM Presentation-style {STYLE_ID_OF(PStyle1)
| STYLE_ID_OF(PStyle2)
| STYLE_ID_OF(PStyle3)},
PERM Resource {ANY_VALUE},
REQ Application-comments {REQ #constraint-name {"29"},
PERM #external-data {ANY_VALUE}}
}

```

SPECIFIC:

```

REQ Object-type           {'block'},
REQ Object-identifier     {ANY_VALUE},
CASE $DAC OF {
  $FDA:
    PERM Object-class     {OBJECT_CLASS_ID_OF(GenericBlock)}
  $FPDA:
    REQ Object-class      {OBJECT_CLASS_ID_OF(GenericBlock)}
},
PERM Presentation-style   {STYLE_ID_OF(PStyle1)
| STYLE_ID_OF(PStyle2)
| STYLE_ID_OF(PStyle3)},
PERM Content-architecture-class {$FC | $FPC | $FPR | $FPG },
CASE GenericBlock (Object-class) OF {
  VOID:
    REQ Content-portions  {CONTENT_ID_OF(Character-content-portion)+
|CONTENT_ID_OF(Raster-graphics-content-portion)
|CONTENT_ID_OF(Geometric-graphics-content-portion)}
},
PERM Presentation-attributes {
  PERM #character-attributes {
    PERM #alignment         {ANY_VALUE},
    PERM #character-fonts   {ANY_VALUE},
    PERM #character-path    {ANY_VALUE},
    PERM #character-spacing {ANY_VALUE},
    PERM #character-orientation {ANY_VALUE},
    PERM #code-extension-announcers {$CDEXTEN},
    PERM #first-line-offset {ANY_VALUE},
    PERM #graphic-character-sets {$PERMIT-GRCHAR},
    PERM #graphic-character-subrepertoire {ANY_VALUE},
    PERM #graphic-rendition {$GRAPHICRENDITIONS},
    PERM #itemization       {ANY_VALUE},
    PERM #kerning-offset    {ANY_VALUE},
    PERM #line-layout-table {ANY_VALUE},
    PERM #line-progression  {ANY_VALUE},
    PERM #line-spacing      {ANY_VALUE},
    PERM #pairwise-kerning  {ANY_VALUE},
    PERM #formatting-indicator {ANY_VALUE},
    PERM #initial-offset    {ANY_VALUE}
  }
},
PERM Application-comments {REQ #constraint-name {"29"},
PERM #external-data{ANY_VALUE}}

```

SPECIFIC_AND_GENERIC:

```

PERM Position             {REQ #fixed-position
|REQ #horizontal-position {ANY_VALUE},
|REQ #vertical-position {ANY_VALUE}}},
PERM Dimensions           {REQ #horizontal-dimension
|REQ #fixed-dimension {ANY_VALUE}},
|REQ #vertical-dimension
|REQ #fixed-dimension {ANY_VALUE}}},
PERM Transparency         {ANY_VALUE},
PERM Colour               {ANY_VALUE},
PERM Border               {ANY_VALUE},
PERM User-readable-comments {ANY_STRING},
PERM User-visible-name    {ANY_STRING}
}

```

7.4.3.33 SpecificBlock

```

{
SPECIFIC:
REQ Object-type           {'block'},
REQ Object-identifier     {ANY_VALUE},
REQ Content-portions      {CONTENT_ID_OF(Character-content-portion)+
|CONTENT_ID_OF(Raster-graphics-content-portion)
|CONTENT_ID_OF(Geometric-graphics-content-portion)},

```

```

PERM Position {REQ #fixed-position
                 {REQ #horizontal-position {ANY_VALUE},
                  REQ #vertical-position {ANY_VALUE}}},
PERM Dimensions {REQ #horizontal-dimension
                   {REQ #fixed-dimension {ANY_VALUE}},
                   REQ #vertical-dimension
                   {REQ #fixed-dimension {ANY_VALUE}}},
PERM Presentation-style {STYLE_ID_OF(PStyle1)
                          |STYLE_ID_OF(PStyle2)
                          |STYLE_ID_OF(PStyle3)
                          |STYLE_ID_OF(PStyle4)},
PERM Content-architecture-class {$FC | $FPC | $FPR | $FPG},
PERM Presentation-attributes {
  PERM #character-attributes {
    PERM #alignment {ANY_VALUE},
    PERM #character-fonts {ANY_VALUE},
    PERM #character-path {ANY_VALUE},
    PERM #character-spacing {ANY_VALUE},
    PERM #character-orientation {ANY_VALUE},
    PERM #code-extension-announcers {$CDEXTEN},
    PERM #first-line-offset {ANY_VALUE},
    PERM #graphic-character-sets {$PERMIT-GRCHAR },
    PERM #graphic-character-subrepertoire {ANY_VALUE},
    PERM #graphic-rendition {$GRAPHICRENDITIONS},
    PERM #itemization {ANY_VALUE},
    PERM #kerning-offset {ANY_VALUE},
    PERM #line-layout-table {ANY_VALUE},
    PERM #line-progression {ANY_VALUE},
    PERM #line-spacing {ANY_VALUE},
    PERM #pairwise-kerning {ANY_VALUE},
    PERM #formatting-indicator {ANY_VALUE},
    PERM #initial-offset {ANY_VALUE}
  }
},
PERM Transparency {ANY_VALUE},
PERM Colour {ANY_VALUE},
PERM Border {ANY_VALUE},
PERM User-readable-comments {ANY_STRING},
PERM User-visible-name {ANY_STRING},
PERM Application-comments {REQ #constraint-name {"30"},
                              PERM #external-data {ANY_VALUE}}
}

```

7.4.3.34 FormArea

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates {$FormAreaGFS}},
    REQ Position {REQ #variable-position {
                  PERM #offset {ANY_VALUE},
                  PERM #separation {ANY_VALUE},
                  PERM #alignment {ANY_VALUE},
                  PERM #fill-order {'normal-order'}}},
    REQ Dimensions {REQ #horizontal-dimension
                   {REQ #fixed-dimension {ANY_VALUE}},
                   REQ #vertical-dimension
                   {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Layout-path {'270-degrees'},
    REQ Application-comments {REQ #constraint-name {"31"},
                              PERM #external-data {ANY_VALUE}}
  }
SPECIFIC:
  CASE $DAC OF {
    $FDA:
      PERM Object-class {OBJECT_CLASS_ID_OF(FormArea)}
    $FPDA:
      REQ Object-class {OBJECT_CLASS_ID_OF(FormArea)}
  }
}

```

```

REQ Subordinates      {SUB_ID_OF(ArrangedContentFixed)+,
                      SUB_ID_OF(FormEntryArea)+,
                      SUB_ID_OF(EntryGroupArea)+},
PERM Position         {ANY_VALUE},
PERM Dimensions       {REQ #horizontal-dimension
                      {REQ #fixed-dimension {ANY_VALUE}},
                      REQ #vertical-dimension
                      {REQ #fixed-dimension {ANY_VALUE}}},
PERM Application-comments {REQ #constraint-name {"31"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM Transparency   {ANY_VALUE},
  PERM Colour         {ANY_VALUE},
  PERM Border         {ANY_VALUE}
}

```

7.4.3.35 EntryGroupArea

```

:ANY-FRAME-FIXED {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates {$EntryGroupAreaGFS}
    },
REQ Position         {REQ #fixed-position
                      {REQ #horizontal-position {ANY_VALUE},
                      REQ #vertical-position {ANY_VALUE}}},
REQ Dimensions       {REQ #horizontal-dimension
                      {REQ #fixed-dimension {ANY_VALUE}},
                      REQ #vertical-dimension
                      {REQ #fixed-dimension {ANY_VALUE}}},
PERM Layout-path     {'270-degrees'},
REQ Application-comments {REQ #constraint-name {"35"},
PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE $DAC OF {
    $FDA:
      PERM Object-class {OBJECT_CLASS_ID_OF(EntryGroupArea)}
    $FPDA:
      REQ Object-class   {OBJECT_CLASS_ID_OF(EntryGroupArea)}
    },
REQ Subordinates      {SUB_ID_OF(ArrangedContentFixed)+,
                      SUB_ID_OF(FormEntryArea)+,
                      SUB_ID_OF(EntryGroupArea)+},
PERM Position         {ANY_VALUE},
PERM Dimensions       {REQ #horizontal-dimension
                      {REQ #fixed-dimension {ANY_VALUE}},
                      REQ #vertical-dimension
                      {REQ #fixed-dimension {ANY_VALUE}}},
PERM Imaging-order    {ANY_VALUE},
PERM Application-comments {REQ #constraint-name {"35"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM Transparency   {ANY_VALUE},
  PERM Colour         {ANY_VALUE},
  PERM Border         {ANY_VALUE}
}

```

7.4.3.36 TableArea

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates {$TableAreaGFS}
    },

```



```

REQ Position {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},

REQ Dimensions {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
    {REQ #rule-b {ANY_VALUE}
    [REQ #fixed-dimension {ANY_VALUE}]},
    {'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)},
    {ANY_VALUE},
PERM Position {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
    {REQ #fixed-dimension {ANY_VALUE}}},
PERM Dimensions {REQ #constraint-name {'36'},
    PERM #external-data {ANY_VALUE}}

PERM Application-comments

SPECIFIC_AND_GENERIC:
PERM Transparency {ANY_VALUE},
PERM Colour {ANY_VALUE},
PERM Border {ANY_VALUE}
}

```

7.4.3.37 TableHeader

```

:ANY-FRAME-VARIABLE {
GENERIC:
CASE $DAC OF {
    $PDA-FPDA:
        REQ Generator-for-subordinates {$TableHeaderGFS}
    },
REQ Position {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},

REQ Dimensions {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
    {REQ #rule-b {ANY_VALUE}
    [REQ #fixed-dimension {ANY_VALUE}]},
    {'270-degrees'},
PERM Layout-path {REQ #constraint-name {'34'},
    PERM #external-data {ANY_VALUE}}

REQ Application-comments

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

```

```

REQ Subordinates      {SUB_ID_OF(SourcedContentFixed)+},
PERM Position         {ANY_VALUE},
PERM Dimensions       {REQ #horizontal-dimension
                      {REQ #fixed-dimension {ANY_VALUE}},
                      REQ #vertical-dimension
                      {REQ #fixed-dimension {ANY_VALUE}}},
PERM Application-comments {REQ #constraint-name {"34"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM Transparency   {ANY_VALUE},
  PERM Colour         {ANY_VALUE},
  PERM Border         {ANY_VALUE}
}

```

7.4.3.38 TableLabel

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates  {$TableLabelGFS}
    },
  REQ Position                       {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},
  REQ Dimensions                     {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #rule-b {ANY_VALUE}
    {REQ #fixed-dimension {ANY_VALUE}}},
  PERM Layout-path                   {'270-degrees'},
  REQ Application-comments           {REQ #constraint-name {"37"},
  PERM #external-data {ANY_VALUE}}

SPECIFIC:
  CASE $DAC OF {
    $FDA:
      PERM Object-class               {OBJECT_CLASS_ID_OF(TableLabel)}
    $FPDA:
      REQ Object-class                 {OBJECT_CLASS_ID_OF(TableLabel)}
    },
  REQ Subordinates                   {SUB_ID_OF(TableLabelContent)+,
  SUB_ID_OF(CompositeTableLabel)},
  PERM Position                       {ANY_VALUE},
  PERM Dimensions                     {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #fixed-dimension {ANY_VALUE}}},
  PERM Application-comments           {REQ #constraint-name {"37"},
  PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
  PERM Transparency                   {ANY_VALUE},
  PERM Colour                         {ANY_VALUE},
  PERM Border                         {ANY_VALUE}
}

```

7.4.3.39 CompositeTableLabel

```

:ANY-FRAME-VARIABLE {
GENERIC:
  CASE $DAC OF {
    $PDA-FPDA:
      REQ Generator-for-subordinates  {$CompositeTableLabelGFS}
    },
  REQ Position                       {REQ #fixed-position {
    REQ #horizontal-position {ANY_VALUE},
    REQ #vertical-position {ANY_VALUE}}},

```

```

REQ Dimensions {REQ #horizontal-dimension
                {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
                {REQ #rule-b {ANY_VALUE}
                [REQ #fixed-dimension {ANY_VALUE}
                [REQ #maximum-size {'applies'}]},
                {'270-degrees'},
PERM Layout-path {REQ #constraint-name {"38"},
REQ Application-comments {PERM #external-data {ANY_VALUE}}

SPECIFIC:
CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF
                      (CompositeTableLabel)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF
                    (CompositeTableLabel)}
},
REQ Subordinates {SUB_ID_OF(LabelComponent)+},
PERM Position {ANY_VALUE},
PERM Dimensions {REQ #horizontal-dimension
                {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
                {REQ #fixed-dimension {ANY_VALUE}}},
PERM Imaging-order {ANY_VALUE},
PERM Application-comments {REQ #constraint-name {"38"},
PERM #external-data {ANY_VALUE}}

SPECIFIC_AND_GENERIC:
PERM Transparency {ANY_VALUE},
PERM Colour {ANY_VALUE},
PERM Border {ANY_VALUE}
}

```

7.4.3.40 LabelComponent

```

:ANY-FRAME-VARIABLE {
GENERIC:
CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates {$LabelComponentGFS}
},
REQ Position {REQ #variable-position {
              PERM #offset {ANY_VALUE},
              PERM #separation {ANY_VALUE},
              PERM #alignment {ANY_VALUE},
              PERM #fill-order {'normal-order'}}},
REQ Dimensions {REQ #horizontal-dimension
                {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
                {REQ #rule-b {ANY_VALUE}
                [REQ #fixed-dimension {ANY_VALUE}
                [REQ #maximum-size {'applies'}]},
                {'270-degrees'},
PERM Layout-path {REQ #constraint-name {"39"},
REQ Application-comments {PERM #external-data {ANY_VALUE}}

SPECIFIC:
CASE $DAC OF {
  $FDA:
    PERM Object-class {OBJECT_CLASS_ID_OF(LabelComponent)}
  $FPDA:
    REQ Object-class {OBJECT_CLASS_ID_OF(LabelComponent)}
},
REQ Subordinates {SUB_ID_OF(TableLabelContent)+},
PERM Position {ANY_VALUE},
PERM Dimensions {REQ #horizontal-dimension
                {REQ #fixed-dimension {ANY_VALUE}},
REQ #vertical-dimension
                {REQ #fixed-dimension {ANY_VALUE}}},

```

```

    PERM Application-comments      {REQ #constraint-name {"39"},
                                  PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency              {ANY_VALUE},
    PERM Colour                    {ANY_VALUE},
    PERM Border                    {ANY_VALUE}
}

7.4.3.41 RowArea

:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ Generator-for-subordinates  {$RowAreaGFS}
        },
    REQ Position                    {REQ #variable-position {
                                  PERM #offset {ANY_VALUE},
                                  PERM #separation {ANY_VALUE},
                                  PERM #alignment {ANY_VALUE},
                                  PERM #fill-order {'normal-order'}}},
    REQ Dimensions                  {REQ #horizontal-dimension
                                  {REQ #fixed-dimension {ANY_VALUE}},
                                  REQ #vertical-dimension
                                  {REQ #rule-b {ANY_VALUE}
                                   [REQ #fixed-dimension {ANY_VALUE}]},
                                  {'270-degrees'},
    PERM Layout-path                {REQ #constraint-name {"40"},
    REQ Application-comments          PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class        {OBJECT_CLASS_ID_OF(RowArea)}
        $FPDA:
            REQ Object-class         {OBJECT_CLASS_ID_OF(RowArea)}
        },
    REQ Subordinates                {SUB_ID_OF(Cell)+,
    SUB_ID_OF(SubRowGroup)},
    PERM Position                  {ANY_VALUE},
    PERM Dimensions                 {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #fixed-dimension {ANY_VALUE}}},
    PERM Application-comments       {REQ #constraint-name {"40"},
    PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Transparency              {ANY_VALUE},
    PERM Colour                    {ANY_VALUE},
    PERM Border                    {ANY_VALUE}
}

```

7.4.3.42 Cell

```

:ANY-FRAME-VARIABLE {
GENERIC:
    REQ Position                    {REQ #fixed-position {
    REQ #horizontal-position {ANY_VALUE},
    REQ #vertical-position {ANY_VALUE}}},
    REQ Dimensions                  {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #rule-b {ANY_VALUE}
    [REQ #fixed-dimension {ANY_VALUE}
    [REQ #maximum-size {'applies'}]}},
    PERM Layout-path                {'270-degrees'},
    REQ Permitted-categories         {ANY_STRING...},
    -- category name for tables should be specified --
    REQ Application-comments         {REQ #constraint-name {"41"},
    PERM #external-data {ANY_VALUE}}
}

```

SPECIFIC:

```
CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF(Cell)}
  $FPDA:
    REQ Object-class      {OBJECT_CLASS_ID_OF(Cell)}
},
REQ Subordinates          {SUB_ID_OF(SpecificBlock)},
PERM Position             {ANY_VALUE},
PERM Dimensions           {REQ #horizontal-dimension
                          {REQ #fixed-dimension {ANY_VALUE}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}}},
PERM Imaging-order        {ANY_VALUE},
PERM Application-comments {REQ #constraint-name {"41"},
                          PERM #external-data {ANY_VALUE}}
```

SPECIFIC_AND_GENERIC:

```
PERM Transparency        {ANY_VALUE},
PERM Colour               {ANY_VALUE},
PERM Border               {ANY_VALUE}
```

}

7.4.3.43 SubRowGroup

:ANY-FRAME-VARIABLE {

GENERIC:

```
CASE $DAC OF {
  $PDA-FPDA:
    REQ Generator-for-subordinates {$SubRowGroupGFS}
},
REQ Position              {REQ #fixed-position {
                          REQ #horizontal-position {ANY_VALUE},
                          REQ #vertical-position {ANY_VALUE}}},
REQ Dimensions            {REQ #horizontal-dimension
                          {REQ #fixed-dimension {ANY_VALUE}},
                          REQ #vertical-dimension
                          {REQ #rule-b {ANY_VALUE}
                          [REQ #fixed-dimension {ANY_VALUE}
                          [REQ #maximum-size {'applies'}]},
                          PERM Layout-path          {'270-degrees'},
                          REQ Application-comments    {REQ #constraint-name {"42"},
                          PERM #external-data {ANY_VALUE}}
```

SPECIFIC:

```
CASE $DAC OF {
  $FDA:
    PERM Object-class      {OBJECT_CLASS_ID_OF(SubRowGroup)}
  $FPDA:
    REQ Object-class      {OBJECT_CLASS_ID_OF(SubRowGroup)}
},
REQ Subordinates          {SUB_ID_OF(SubRow)+},
PERM Position             {ANY_VALUE},
PERM Dimensions           {REQ #horizontal-dimension
                          {REQ #fixed-dimension {ANY_VALUE}},
                          REQ #vertical-dimension
                          {REQ #fixed-dimension {ANY_VALUE}}},
PERM Application-comments {REQ #constraint-name {"42"},
                          PERM #external-data {ANY_VALUE}}
```

SPECIFIC_AND_GENERIC:

```
PERM Transparency        {ANY_VALUE},
PERM Colour               {ANY_VALUE},
PERM Border               {ANY_VALUE}
```

}

7.4.3.44 SubRow

:ANY-FRAME-VARIABLE {

GENERIC:

```

CASE $DAC OF {
    $PDA-FPDA:
        REQ Generator-for-subordinates {SubRowGFS}
    },
REQ Position {REQ #variable-position {
    PERM #offset {ANY_VALUE},
    PERM #separation {ANY_VALUE},
    PERM #alignment {ANY_VALUE},
    PERM #fill-order {'normal-order'}}},
REQ Dimensions {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #rule-b {ANY_VALUE}
    |REQ #fixed-dimension {ANY_VALUE}
    |REQ #maximum-size {'applies'}}},
PERM Layout-path {'270-degrees'},
REQ Application-comments {REQ #constraint-name {"43"},
    PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
    $FDA:
        PERM Object-class {OBJECT_CLASS_ID_OF(SubRow)}
    $FPDA:
        REQ Object-class {OBJECT_CLASS_ID_OF(SubRow)}
    },
REQ Subordinates {SUB_ID_OF(Cell)+},
PERM Position {ANY_VALUE},
PERM Dimensions {REQ #horizontal-dimension
    {REQ #fixed-dimension {ANY_VALUE}},
    REQ #vertical-dimension
    {REQ #fixed-dimension {ANY_VALUE}}},
PERM Application-comments {REQ #constraint-name {"43"},
    PERM #external-data {ANY_VALUE}}

```

SPECIFIC_AND_GENERIC:

```

PERM Transparency {ANY_VALUE},
PERM Colour {ANY_VALUE},
PERM Border {ANY_VALUE}

```

}

7.4.3.45 TableLabelContent

:ANY-FRAME-VARIABLE {

GENERIC:

```

CASE $DAC OF {
    $PDA-FPDA:
        REQ Logical-source {OBJECT_CLASS_ID_OF(CommonContent)},
        REQ Position {REQ #fixed-position
            {REQ #horizontal-position {ANY_VALUE},
            REQ #vertical-position {ANY_VALUE}}},
        REQ Dimensions {REQ #horizontal-dimension
            {REQ #fixed-dimension {ANY_VALUE}},
            REQ #vertical-dimension
            {REQ #fixed-dimension {ANY_VALUE}
            |REQ #maximum-size {'applies'}}},
        PERM Layout-path {'270-degrees'}
    },
REQ Application-comments {REQ #constraint-name {"44"},
    PERM #external-data {ANY_VALUE}}

```

SPECIFIC:

```

CASE $DAC OF {
    $FDA:
        PERM Object-class {OBJECT_CLASS_ID_OF
            (TableLabelContent)}

```

```

    $FPDA:
        REQ Object-class      {OBJECT_CLASS_ID_OF
                               (TableLabelContent)}
    },
    REQ Subordinates         {SUB_ID_OF(SpecificBlock)+},
    PERM Application-comments {REQ #constraint-name {"44"},
                               PERM #external-data {ANY_VALUE}}
}

7.4.3.46 FormEntryArea

:ANY-FRAME-VARIABLE {
GENERIC:
    CASE $DAC OF {
        $PDA-FPDA:
            REQ Position      {REQ #fixed-position
                               {REQ #horizontal-position {ANY_VALUE},
                               REQ #vertical-position {ANY_VALUE}}},
            REQ Dimensions    {REQ #horizontal-dimension
                               {REQ #fixed-dimension {ANY_VALUE}},
                               REQ #vertical-dimension
                               {REQ #fixed-dimension {ANY_VALUE}}},
            PERM Layout-path  {'270-degrees'}
    },
    REQ Application-comments {REQ #constraint-name {"45"},
                               PERM #external-data {ANY_VALUE}}
SPECIFIC:
    CASE $DAC OF {
        $FDA:
            PERM Object-class {OBJECT_CLASS_ID_OF(FormEntryArea)}
        $FPDA:
            REQ Object-class  {OBJECT_CLASS_ID_OF(FormEntryArea)}
    },
    REQ Subordinates         {SUB_ID_OF(SpecificBlock)},
    PERM Application-comments {REQ #constraint-name {"45"},
                               PERM #external-data {ANY_VALUE}}
SPECIFIC_AND_GENERIC:
    PERM Permitted-categories {ANY_STRING...}
}

```

7.5 Layout style constituent constraints

7.5.1 Macro definitions

```

DEFINE(SameLayoutObject,"
REQ    {REQ #logical-object {<object-id-expr>::=PREC-OBJ(CURR-OBJ);}
|REQ #logical-object {'null'}},
PERM #layout-object {'page'      -- to layout object type --
| ANY_STRING           -- to layout category --
| OBJECT_CLASS_ID_OF(ColumnFixed)
| OBJECT_CLASS_ID_OF(ColumnVariable)
| OBJECT_CLASS_ID_OF(CompositeColumnFixed)
| OBJECT_CLASS_ID_OF(CompositeColumnVariable)}
")

```

7.5.2 Factor constraints

```

FACTOR ANY-LAYOUT-STYLE
{
REQ    Layout-style-identifier {ANY_VALUE},
PERM   User-readable-comments  {ANY_STRING},
PERM   User-visible-name       {ANY_STRING}
}

```

7.5.3 Constituent constraints

7.5.3.1 LStyle1

:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints *Passage, NumberedSegment, Title, Caption, Paragraph, Phrase, Footnote, Figure, Reference, Description, NumberedList, UnNumberedList, DefinitionList, ListItem, ListTerm* --

CASE \$GLAS OF{

\$COMPLETE:

PERM **Indivisibility** {ANY_VALUE},
PERM **Layout-object-class** {ANY_VALUE},
PERM **New-layout-object** {ANY_VALUE},
PERM **Same-layout-object** {\$SameLayoutObject},
PERM **Synchronization** {ANY_VALUE}

VOID:

PERM **Indivisibility** {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
PERM **New-layout-object** {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
PERM **Same-layout-object** {REQ {REQ #logical-object
{<object-id-expr>::=PREC-Obj(Curr-Obj);
|REQ #logical-object {'null'}},
PERM #layout-object
{ANY_STRING --to layout category--
|'page' --to layout object type--
},
PERM **Synchronization** {ANY_VALUE}

}}

7.5.3.2 LStyle2

:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints *BodyText and Number* --

CASE \$GLAS OF{

\$COMPLETE:

PERM **Block-alignment** {ANY_VALUE},
PERM **Concatenation** {ANY_VALUE},
PERM **Indivisibility** {ANY_VALUE},
PERM **Layout-category** {ANY_VALUE},
PERM **Layout-object-class** {ANY_VALUE},
PERM **New-layout-object** {ANY_VALUE},
PERM **Offset** {ANY_VALUE},
PERM **Same-layout-object** {\$SameLayoutObject},
PERM **Separation** {ANY_VALUE},
PERM **Synchronization** {ANY_VALUE}

VOID:

PERM **Block-alignment** {ANY_VALUE},
PERM **Concatenation** {ANY_VALUE},
PERM **Indivisibility** {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
PERM **Layout-category** {ANY_VALUE},
PERM **New-layout-object** {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
PERM **Offset** {ANY_VALUE},
PERM **Same-layout-object** {REQ {REQ #logical-object
{<object-id-expr>::=PREC-Obj(Curr-Obj);
|REQ #logical-object {'null'}},
PERM #layout-object
{ANY_STRING --to layout category--
|'page' --to layout object type--
},
PERM **Separation** {ANY_VALUE},
PERM **Synchronization** {ANY_VALUE}

}}

7.5.3.3 LStyle3

```
:ANY-LAYOUT-STYLE{
-- This style is used for the constituent constraints CommonText, PageNumber, TableNumber, CurrentInstance,
CommonNumber and CommonReference. --
PERM   Block-alignment           {ANY_VALUE},
PERM   Concatenation             {ANY_VALUE},
PERM   Offset                   {ANY_VALUE},
PERM   Separation               {ANY_VALUE}
}
```

7.5.3.4 LStyle4

-- This style is not used --

7.5.3.5 LStyle5

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints BodyRaster and BodyGeometric. --
```

```
CASE $GLAS OF{
$COMPLETE:
  PERM   Block-alignment           {ANY_VALUE},
  PERM   Layout-category          {ANY_VALUE},
  PERM   Layout-object-class      {ANY_VALUE},
  PERM   New-layout-object        {ANY_VALUE},
  PERM   Offset                   {ANY_VALUE},
  PERM   Same-layout-object       {$SameLayoutObject},
  PERM   Separation               {ANY_VALUE},
  PERM   Synchronization         {ANY_VALUE}
VOID:
  PERM   Block-alignment           {ANY_VALUE},
  PERM   Layout-category          {ANY_VALUE},
  PERM   New-layout-object        {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
  PERM   Offset                   {ANY_VALUE},
  PERM   Same-layout-object       {REQ {REQ #logical-object
|<object-id-expr>::=PREC-OBJ(CURR-OBJ);
|REQ #logical-object {'null'}},
PERM #layout-object
|ANY_STRING --to layout category--
|'page' --to layout object type--
},
  PERM   Separation               {ANY_VALUE},
  PERM   Synchronization         {ANY_VALUE}
}}
```

7.5.3.6 LStyle6

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint FootnoteText. --
```

```
CASE $GLAS OF{
$COMPLETE:
  PERM   Indivisibility           {ANY_VALUE},
  PERM   Block-alignment           {ANY_VALUE},
  PERM   Concatenation             {ANY_VALUE},
  REQ    Layout-category          {$FOOTNOTECATEGORY},
  PERM   Offset                   {ANY_VALUE},
  PERM   Separation               {ANY_VALUE}
VOID:
  PERM   Indivisibility           {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
  PERM   Block-alignment           {ANY_VALUE},
  PERM   Concatenation             {ANY_VALUE},
  REQ    Layout-category          {$FOOTNOTECATEGORY},
  PERM   Offset                   {ANY_VALUE},
  PERM   Separation               {ANY_VALUE}
}}
```

7.5.3.7 LStyle7

-- This style is not used. --

7.5.3.8 LStyle8

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints CommonRaster and CommonGeometric. --
PERM   Block-alignment           {ANY_VALUE},
PERM   Offset                     {ANY_VALUE},
PERM   Separation                 {ANY_VALUE}
}
```

7.5.3.9 LStyle9

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint FootnoteNumber. --
PERM   Block-alignment           {ANY_VALUE},
REQ    Layout-category           {$FOOTNOTECATEGORY},
PERM   Offset                     {ANY_VALUE},
PERM   Separation                 {ANY_VALUE}
}
```

7.5.3.10 LStyle10

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints FootnoteReference and ReferencedContent. --
CASE $GLAS OF{
  $COMPLETE:
    PERM   Block-alignment           {ANY_VALUE},
    PERM   Concatenation             {ANY_VALUE},
    PERM   Indivisibility           {ANY_VALUE},
    PERM   Layout-category           {ANY_VALUE},
    PERM   Offset                     {ANY_VALUE},
    PERM   Same-layout-object       {$SameLayoutObject},
    PERM   Separation                 {ANY_VALUE}
  VOID:
    PERM   Block-alignment           {ANY_VALUE},
    PERM   Concatenation             {ANY_VALUE},
    PERM   Indivisibility           {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
    PERM   Layout-category           {ANY_VALUE},
    PERM   Offset                     {ANY_VALUE},
    PERM   Same-layout-object       {REQ {REQ #logical-object
|<object-id-expr>::=PREC-Obj(CURR-Obj);
| REQ #logical-object {'null'}},
PERM #layout-object
|ANY_STRING --to layout category--
|'page' --to layout object type--
},
    PERM   Separation                 {ANY_VALUE}
}}
}
```

7.5.3.11 LStyle11

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint FootnoteBody. --
CASE $GLAS OF{
  $COMPLETE:
    PERM   Indivisibility           {ANY_VALUE},
    PERM   Same-layout-object       {$SameLayoutObject},
    PERM   Synchronization         {ANY_VALUE}
  VOID:
    PERM   Indivisibility           {ANY_STRING --to layout category--
|'page' --to layout object type--
|'null'},
}
```

```

    PERM    Same-layout-object          {REQ {REQ #logical-object
                                         {<object-id-expr>:=PREC-Obj(CURR-Obj);}
                                         | REQ #logical-object {'null'}},
    PERM    #layout-object              {ANY_STRING --to layout category--
                                         |'page' --to layout object type--
                                         },
    PERM    Synchronization            {ANY_VALUE}
}}

```

7.5.3.12 LStyle12

```

:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint Artwork. --
CASE $GLAS OF{
$COMPLETE:
    PERM    Indivisibility              {ANY_STRING -- to layout category --
                                         |'page' -- to layout object type --
                                         |'null'},
    PERM    Layout-object-class         {OBJECT_CLASS_ID_OF(CompositeArtwork)},
    PERM    New-layout-object           {OBJECT_CLASS_ID_OF(CompositeColumnFixed)
                                         |OBJECT_CLASS_ID_OF(CompositeColumnVariable)
                                         -- to layout object class --
                                         |ANY_STRING -- to layout category --
                                         |'page' -- to layout object type --
                                         |'null'},
    PERM    Synchronization            {ANY_VALUE}
VOID:
    PERM    Indivisibility              {ANY_STRING --to layout category--
                                         |'page' --to layout object type--
                                         |'null'},
    PERM    New-layout-object           {ANY_STRING --to layout category--
                                         |'page' --to layout object type--
                                         |'null'},
    PERM    Synchronization            {ANY_VALUE}
}}

```

7.5.3.13 LStyleT1

```

:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint Form. --
REQ    Layout-object-class            {OBJECT_CLASS_ID_OF(FormArea)}
}

```

7.5.3.14 LStyleT2

```

ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint EntryElement. --

-- In the case of Form, the following attribute shall be specified. --
REQ    Layout-object-class            {OBJECT_CLASS_ID_OF(FormEntryArea)}
|
-- In the case of Table, both of the following attributes shall be specified. --
REQ    New-layout-object              {OBJECT_CLASS_ID_OF(Cell) -- to layout object class --
                                         |ANY_STRING -- to layout category --},
REQ    Indivisibility                 {OBJECT_CLASS_ID_OF(Cell) -- to layout object class --
                                         |ANY_STRING -- to layout category -- }
}

```

7.5.3.15 LStyleT3

```

:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint EntryGroup. --
REQ    Layout-object-class            {OBJECT_CLASS_ID_OF(EntryGroupArea)}
}

```

7.5.3.16 LStyleT4

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint Table. --
REQ    New-layout-object      {OBJECT_CLASS_ID_OF(TableArea)},
PERM   Indivisibility         {OBJECT_CLASS_ID_OF(TableArea)
                                -- to layout object class --
                                |ANY_STRING -- to layout category --
                                |'page'    -- to layout object type --
                                |'null' },
PERM   Same-layout-object     {$SameLayoutObject}
}
```

7.5.3.17 LStyleT5

```
:ANY-LAYOUT-STYLE {
-- This layout style is used for the constituent constraint Row. --
REQ    New-layout-object      {OBJECT_CLASS_ID_OF(RowArea)},
PERM   Indivisibility         {OBJECT_CLASS_ID_OF(RowArea)
                                -- to layout object class --
                                |ANY_STRING -- to layout category --
                                |'page'    -- to layout object type --
                                |'null' }
}
```

7.5.3.18 LStyleT6

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint TableComponent. --
REQ    Layout-object-class    {OBJECT_CLASS_ID_OF(SubRowGroup)}
}
```

7.5.3.19 LStyleT7

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint RowComponent. --
REQ    New-layout-object      {OBJECT_CLASS_ID_OF(SubRow)},
PERM   Indivisibility         {OBJECT_CLASS_ID_OF(SubRow)
                                -- to layout object class --
                                |ANY_STRING -- to layout category --
                                |'null' }
}
```

7.5.3.20 LStyleT8

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraint Table. --
PERM   Indivisibility         {OBJECT_CLASS_ID_OF(TableArea)
                                -- to layout object class --
                                |ANY_STRING -- to layout category --
                                |'page'    -- to layout object type --
                                |'null' },
PERM   Same-layout-object     {$SameLayoutObject}
}
```

7.5.3.21 LStyleT9

```
:ANY-LAYOUT-STYLE {
-- This style is used for the constituent constraints EntryText, EntryRaster and EntryGeometric. --
PERM   Block-alignment        {ANY_VALUE},
PERM   Layout-category        {ANY_VALUE},
PERM   Offset                  {ANY_VALUE}
}
```

7.6 Presentation style constituent constraints

7.6.1 Macro definitions

-- No macro definitions are applicable to this subclause. --

7.6.2 Factor constraints

FACTOR ANY-PRESENTATION-STYLE

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

7.6.3 Constituent constraints

7.6.3.1 PStyle1

:ANY-PRESENTATION-STYLE {

-- This style is used for the constituent constraints *BodyText*, *Number*, *FootnoteNumber*, *FootnoteReference*, *FootnoteText*, *EntryText*, *ReferencedContent*, *GenericBlock* and *SpecificBlock*. --

PERM Presentation-attributes {

```
  PERM #character-attributes {
    PERM #alignment            {ANY_VALUE},
    PERM #character-fonts      {ANY_VALUE},
    PERM #character-orientation {ANY_VALUE},
    PERM #character-path       {ANY_VALUE},
    PERM #character-spacing    {ANY_VALUE},
    PERM #code-extension-announcers {$CDEXTEN},
    PERM #first-line-offset     {ANY_VALUE},
    PERM #graphic-character-sets {$PERMIT-GRCHAR},
    PERM #graphic-character-subrepertoire {ANY_VALUE},
    PERM #graphic-rendition     {$GRAPHICRENDITIONS},
    PERM #indentation          {ANY_VALUE},
    PERM #itemization           {ANY_VALUE},
    PERM #kerning-offset        {ANY_VALUE},
    PERM #line-layout-table     {ANY_VALUE},
    PERM #line-progression      {ANY_VALUE},
    PERM #line-spacing          {ANY_VALUE},
    PERM #orphan-size           {ANY_VALUE},
    PERM #pairwise-kerning      {ANY_VALUE},
    PERM #proportional-line-spacing {ANY_VALUE},
    PERM #widow-size            {ANY_VALUE},
    PERM #initial-offset        {ANY_VALUE},
    PERM #formatting-indicator  {ANY_VALUE}
  }
}}
```

7.6.3.2 PStyle2

:ANY-PRESENTATION-STYLE {

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

```
PERM Presentation-attributes {
  PERM #geometric-graphics-attributes {ANY_VALUE}
}
```

7.6.3.3 PStyle3

```
:ANY-PRESENTATION-STYLE {
-- This style is used for the constituent constraints BodyRaster, CommonRaster, EntryRaster, GenericBlock and
  SpecificBlock. --
PERM Presentation-attributes {
  PERM #raster-graphics-attributes {
    PERM #pel-path {ANY_VALUE},
    PERM #line-progression {ANY_VALUE},
    PERM #pel-spacing {ANY_VALUE},
    PERM #spacing-ratio {ANY_VALUE},
    PERM #clipping {ANY_VALUE},
    PERM #image-dimensions {ANY_VALUE}
  }
}
```

7.6.3.4 PStyle4

```
:ANY-PRESENTATION-STYLE {
-- This style is used for the constituent constraints CommonText, PageNumber, TableNumber, CommonReference,
  CurrentInstance and SpecificBlock. --
PERM Presentation-attributes {
  PERM #character-attributes {
    PERM #alignment {ANY_VALUE},
    PERM #character-fonts {ANY_VALUE},
    PERM #character-orientation {ANY_VALUE},
    PERM #character-path {ANY_VALUE},
    PERM #character-spacing {ANY_VALUE},
    PERM #code-extension-announcers {$CDEXTEN},
    PERM #first-line-offset {ANY_VALUE},
    PERM #graphic-character-sets {$PERMIT-GRCHAR},
    PERM #graphic-character-subrepertoire {ANY_VALUE},
    PERM #graphic-rendition {$GRAPHICRENDITIONS},
    PERM #indentation {ANY_VALUE},
    PERM #itemization {ANY_VALUE},
    PERM #kerning-offset {ANY_VALUE},
    PERM #line-layout-table {ANY_VALUE},
    PERM #line-progression {ANY_VALUE},
    PERM #line-spacing {ANY_VALUE},
    PERM #pairwise-kerning {ANY_VALUE},
    PERM #proportional-line-spacing {ANY_VALUE},
    PERM #initial-offset {ANY_VALUE},
    PERM #formatting-indicator {ANY_VALUE}
  }
}
```

7.7 Content portion constituent constraints

7.7.1 Macro definitions

```
DEFINE(T6, "ASN.1 {2 8 3 7 0}")
DEFINE(T41D, "ASN.1 {2 8 3 7 1}")
DEFINE(T42D, "ASN.1 {2 8 3 7 2}")
DEFINE(Bitmap, "ASN.1 {2 8 3 7 3}")
```

7.7.2 Factor constraints

```
FACTOR ANY-CONTENT {
CASE $DAC OF {
  $FDA :
    REQ Content-identifier-layout {ANY_VALUE}

  $PDA :
    REQ Content-identifier-logical {ANY_VALUE}
    -- This attribute is specified, if the content portion is associated with a basic logical object or a basic
      logical object class. --
}
```

```

|REQ Content-identifier-layout {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic layout object class. --
$FPDA :
|REQ Content-identifier-layout {ANY_VALUE},
|REQ Content-identifier-logical {ANY_VALUE}
-- Both attributes are specified, if the content portion is associated with a basic logical object and a
basic layout object. --
|REQ Content-identifier-layout {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic layout object class. --
|REQ Content-identifier-logical {ANY_VALUE}
-- This attribute is specified, if the content portion is associated with a basic logical object class. --
},
PERM Alternative-representation {ANY_STRING}
}

```

7.7.3 Constituent constraints

7.7.3.1 Character-content-portion

```

:ANY-CONTENT {
PERM Type-of-coding {ASN.1{2 8 3 6 0}},
PERM Content-information {CHARACTER
{
-- Shared Control Functions --
#CR
|#GCC {ANY_VALUE}
|#IGS {ANY_VALUE}
|#LF
|#PLD
|#PLU
|#SCS {ANY_VALUE}
|#SGR {$GRAPHICRENDITIONS}
|#SHS {ANY_VALUE}
|#SLS {ANY_VALUE}
|#SRS {ANY_VALUE}
|#STAB {ANY_VALUE}
|#SUB
|#SVS {ANY_VALUE}
|#VPB {ANY_VALUE}
|#VPR {ANY_VALUE}

-- Layout Control Functions --
|#HPB {ANY_VALUE}
|#HPR {ANY_VALUE}
|#JFY {0}
|#SACS {ANY_VALUE}
|#SRCS {ANY_VALUE}
|#SSW {ANY_VALUE}

-- Logical Control Functions --
|#BPH
|#NBH
|#PTX {ANY_VALUE}

-- Delimiter Functions --
|#SOS
|#ST

-- Space --
|#SP

-- Code extension control functions --
|#LS0
|#LS1R
|#LS2R
|#LS3R

```

```

|#SS2
|#SS3
|#ESC {$DEG-CORE-G0}
|#ESC {$DEG-646-G0}
|#ESC {$DEG-ANY-G1}
|#ESC {$DEG-ANY-G2}
|#ESC {$DEG-ANY-G3}
|#ESC {$DEG-EMPTY-G1}
}...}
}

```

7.7.3.2 Raster-graphics-content-portion

```

:ANY-CONTENT {
PERM  Type-of-coding          {$T6|T41D|T42D|Bitmap},
PERM  Coding-attributes      {
PERM  #raster-graphics-coding-attributes {
    PERM #compression          {ANY_VALUE},
    PERM #number-of-lines      {>0},
    REQ #number-of-pels-per-line {>=0}
  }
},
PERM  Content-information    {RASTER}
}

```

7.7.3.3 Geometric-graphics-content-portion

```

:ANY-CONTENT {
PERM  Type-of-coding          {ASN.1 {2 8 3 8 0}},
PERM  Content-information    {GEOMETRIC}
}

```

8 Format d'échange

Afin de pouvoir se conformer à la présente Recommandation, on doit utiliser le format d'échange de classe A à chaque fois que l'on utilisera le format d'échange ODIF.

NOTE – Le format d'échange SDIF ne s'applique qu'au profil normalisé international.

8.1 Format d'échange de documents de classe A

8.1.1 Format d'échange

La valeur de l'attribut de profil de document «interchange format» pour le présent format d'échange est «if-a». Cette forme d'ODIF est définie dans la Rec. T.415 du CCITT | ISO 8613-5.

8.1.2 Identificateur d'objet du profil d'application de document

La valeur de l'attribut de profil de document «document application profile» pour le présent format d'échange est représentée par l'identificateur d'objet suivant:

```
ASN.1 { 2 8 4 0 36 0 }
```

8.1.3 Codage des commentaires d'application

Le codage de l'attribut «application comments» est défini comme une chaîne d'octets dans la Rec. T.415 du CCITT | ISO 8613-5. Le présent profil d'application de document exige que le codage à l'intérieur de cette chaîne d'octets soit conforme à la syntaxe ASN.1 spécifiée dans la définition de module suivante:

```

FOD_DAPSpecification
DEFINITIONS ::= BEGIN
EXPORTS Appl-Comm-Encoding;

Appl-Comm-Encoding ::= SEQUENCE {
    constraint-nameE [0] IMPLICIT PrintableString OPTIONAL,
    external-data    [1] IMPLICIT OCTET STRING OPTIONAL }
END

```


8.1.4 Longueur des valeurs de données

La longueur maximale des valeurs de données du type OCTET STRING (défini dans la Rec. X.208 du CCITT | ISO 8824 dans des trains de données, pouvant être codée conformément au présent profil d'application de document, est de 32 767 octets. S'il est nécessaire de coder une chaîne de valeurs de données d'une longueur supérieure à celle-ci, le codage de type construit doit être utilisé, c'est à dire que des valeurs de données supérieures à 32 767 en longueur doivent être scindées en une séquence de chaînes inférieures à 32 767, dont chacune est codée en utilisant un type primitif.

Annexe A

Modifications et corrigenda techniques

(Cette annexe fait partie intégrante de la présente Recommandation)

A.1 Modifications

A.1.1 Modifications de la norme de base

Les modifications applicables à la présente Recommandation comprennent un texte à inclure dans la Rec. T.411 du CCITT | ISO 8613-1 en annexes comme suit:

- *Annexe E*: Utilisation de la Norme ISO/CEI 10021 (MOTIS) pour l'échange de documents conformément aux Rec. de la série T.410 du CCITT | ISO 8613 publiées sous la forme d'un premier complément aux Rec. de la série T.410 du CCITT (1988), Addendum II | ISO 8613-1 (1989), Amendement 1.
- *Annexe F*: Proforma et notation correspondante du profil d'application de document publié dans la Rec. T.411 en Annexe F (1991) | ISO 8613-1 (1989), Addendum 1.
- *Annexe G*: Méthodologie pour les essais de conformité. Publié dans l'ISO 8613-1 (1989), Amendement 2.
- *Annexe H*: Enregistrement de documents conformes aux Rec. de la série T.410 du CCITT | ISO 8613 sur des cartouches à disquettes conformes à l'ISO 9293 – Actuellement projet d'Amendement à l'ISO 8613-1 (1989), Annexe H.

La présente Recommandation n'inclut pas les points suivants concernant les modifications:

- Addendum sur la sécurité – Publié en premier complément des Rec. de la série T.410 du CCITT (1988) Addendum IV | ISO 8613 (1989), Addendum 4.
- Addendum sur les styles – Publié en premier complément des Recommandations de la série T.410 du CCITT (1988) Addendum IV | ISO 8613 (1989), Addendum 6.
- Addendum sur l'autre présentation – Publié en premier complément des Rec. de la série T.410 du CCITT (1988) Addendum III | ISO 8613 (1989), Addendum 3.
- Addendum sur la couleur – Publié en révision des Rec. de la série T.410 du CCITT dans le COM III-R 30 – Addendum 2 | ISO 8613 (1989), Amendement 2.
- Addendum sur le contenu graphique par quadrillage de points – Publié en premier complément des Rec. de la série T.410 du CCITT (1988) Addendum I | ISO 8613 (1989), Addendum 1.
- Addendum sur les trains de données – Publié dans l'ISO 8613 (1989), Amendement 5.

De plus, cet amendement traite de l'insertion des Rec. de la série T.410 du CCITT (1988) | ISO 8613 (1988) Corrigenda techniques 1, 2 et 3.

A.1.2 Proposition de modification des normes visant à corriger certains défauts

Cet Amendement traite de l'insertion des Rec. de la série T.410 du CCITT (1988) | ISO 8613 (1988) Corrigenda techniques 1, 2 et 3.

A.2 Corrigenda technique

A.2.1 Corrigenda technique à la présente Recommandation

Néant.

A.3 Versions de la Recommandation ODA

La version de la Recommandation ODA définie en A.1 est connue sous la désignation «version 1.1», sa date officielle est 01.01.1992. La version de la Recommandation ODA contenant tous les amendements, addenda et corrigenda techniques du A.1 est connue sous la désignation «version 1.2» sa date officielle est 01.07.1992.

Les valeurs de l'attribut «version ODA» associée au présent profil d'application de document est:

- *Norme ou Recommandation:*
Rec. de la série T.410 du CCITT (1988) | ISO 8613 (1989); version 1.1.
- *Date de publication:*
01.01.1992.

Annexe B

Méthodes conseillées

(Cette annexe ne fait pas partie intégrante de la présente Recommandation)

B.1 Méthodes de transfert pour l'ODA

B.1.1 Acheminement de l'ODA dans les systèmes conformes à la Recommandation X.400 du CCITT (1984)

La présente méthode conseillée décrit comment des parties de corps ODA sont codées pour être transmises dans le cadre d'un service X.400 du CCITT (1984).

Une partie de corps ODA est codée sous la forme d'une OdaBodyPart dans la définition ci-après:

```
OdaBodyPart ::= SEQUENCE { OdaBodyPartParameters, OdaData }
OdaBodyPartParameters ::= SET {
    document-application-profile
        [0] IMPLICIT OBJECT IDENTIFIER,
    document-architecture-class
        [1] IMPLICIT INTEGER {
            formatted (0),
            processable (1),
            formatted-processable (2) }
OdaData ::= SEQUENCE OF Interchange-Data-Element
```

NOTE – Il est conseillé de transférer un document ODA comme une simple partie de corps avec l'étiquette 12:

```
Oda [12] IMPLICIT OCTETSTRING
```

Le contenu de la chaîne d'octets est codé sous OdaBodyPart, défini ci-dessous. Cependant, cela est hors du champ d'application de la présente Recommandation.

B.1.2 Acheminement de l'ODA sur le FTAM

La présente méthode conseillée décrit le type de document FTAM à utiliser pour le stockage minimal et les capacités de transfert de trains de données ODA. Il est admis que les capacités améliorées peuvent être ajoutées en un point donné.

Lorsqu'il utilise le FTAM pour le transfert d'un fichier ODA, le type de document FTAM-3, «ISO FTAM binaire non structuré» doit être spécifié.

Toutefois, comme les fichiers qui ne contiennent pas de trains de données ODA, peuvent avoir le même type de document, c'est à l'utilisateur des programmes d'application qui accèdent à distance à des fichiers en utilisant le FTAM de savoir qu'un fichier donné contient des trains de données ODA.

B.1.3 Acheminement de l'ODA sur le DTAM

La présente méthode conseillée donne des informations concernant l'échange de document de type ODA avec les protocoles DTAM.

Le DTAM (Transfert et manipulation de documents) est défini dans les Recommandations de la série T.430 et tout comme l'ODA il fait partie intégrante des Recommandations de la série T.400.

Les Recommandations de la série T.520 décrivent les profils d'application de la communication (CAP, *communication application profile*). La Recommandation T.522 décrit le profil d'application de la communication BT1 pour le transfert de masse de document. La Recommandation T.522 est applicable au profil de format de document ouvert (FOD) publié dans la présente Recommandation.

NOTE – L'utilisation du BT1 dans le cadre de services télématiques de bout en bout Telefax 4 et Télétex est décrit en 7.1/T.561 et en 7.1/T.562.

B.1.4 Acheminement de l'ODA sur disquette

La méthode conseillée pour l'échange de documents ODA entre systèmes au moyen de cartouches à disquettes est présentée à l'Annexe H de la Rec. T.411 du CCITT | ISO 8613-1. La présente annexe traite de l'enregistrement de chaque document ODA sous la forme d'un fichier distinct défini par l'ISO 9293.

B.2 Référence aux fontes

La méthode préconisée pour spécifier la fonte de référence s'appuie sur la norme ISO 9541.

Des tailles de 6 à 72 points (100 à 1200 BMU) peuvent être admises dans les réalisations conformes à la présente méthode. Les autres tailles de fontes peuvent être admises, mais dans ce cas, les réalisations doivent disposer d'un mécanisme de repli.

Les propriétés minimales et les valeurs minimales des fontes indiquées dans l'ISO 9541 qui doivent être spécifiées dans un Font-Attribute-Set sont celles spécifiées par la notation de profil d'application de document ci-après:

```

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}
    },
    -- Unités de mesure de base équivalent aux tailles comprises entre 6 et 72 points--
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 Contraintes imposées par la norme ISO 8632 (CGM) dans le cadre du présent profil d'application de document

Dans la pratique, on recommande que l'information de contenu graphique géométrique ne contienne que les éléments indiqués dans la présente partie du document, outre les contraintes imposées par la Rec. T.418 du CCITT | ISO 8613-8. On estime que ce sous-ensemble du CGM est suffisamment utilisé pour permettre l'interfonctionnement des graphiques géométriques pour les applications conformes au présent profil d'application de document.

Lorsqu'un élément est associé à des paramètres, les contraintes sur les valeurs qui font partie de la méthode conseillée sont précisées. Le symbole «- -» indique qu'aucune contrainte n'est imposée dans le cadre de la méthode conseillée.

Il faut respecter les conditions imposées aux éléments obligatoires par l'ISO 8632 et la Rec. T.418 du CCITT | ISO 8613-8.

B.3.1 Eléments délimiteurs

Non-Op	Séquence arbitraire de n octets, où $n = 0, 1, \dots, 32\ 767$. Une séquence de zéro ou plusieurs octets est utilisée pour le remplissage.
Début métafichier	La prise en charge sera assurée pour des chaînes de 254 octets au plus, sauf pour les enregistrements de données qui accepteront des chaînes de longueur maximale égale à 32 767 octets.
Fin métafichier	
Début image	La prise en charge sera assurée pour des chaînes de 254 octets au plus, sauf pour les enregistrements de données qui accepteront des chaînes de longueur maximale égale à 32 767 octets.
Début corps d'image	
Fin image	

B.3.2 Eléments descripteurs de métafichier

Version métafichier	1
Description métafichier	La prise en charge sera assurée pour des chaînes de 254 octets au plus, sauf pour les enregistrements de données de longueur maximale égale à 32 767 octets. Le paramètre de chaîne METAFILE DESCRIPTION sera utilisé pour inclure la sous-chaîne «UIT-T Rec. PM-36» pour étiqueter l'information de contenu conformément à la présente Recommandation. En outre, il est conseillé d'utiliser des générateurs de contenu associés à une sous-chaîne qui identifie la société qui a produit le CGM.
Type VDC	--
Précision des entiers	16, 32
Précision des réels	(0, 9, 23), (1, 16, 16)
Précision d'index	16
Précision de couleur	8, 16
Précision d'index de couleur	8, 16

Index couleur maximum	0, 63
Etendue valeur de couleur	--
Liste d'élément métafichier	--
Remplacement par défaut métafichier	--
Liste de polices	Toutes les polices indiquées dans le métafichier doivent être définies. La désignation de police dans FONT LISTS en utilisant l'ISO 9541 est préférable; mais des noms de police spécifiques peuvent être utilisés.
Liste de polices de caractères	Tous les jeux de caractères référencés dans le métafichier doivent être définis dans la CHARACTER SET LIST. Les jeux de caractères qui sont autorisés sont les mêmes que pour l'architecture de contenu caractères.
Annonceur de codage caractères	--

B.3.3 Eléments descripteurs d'image

Mode échelle	Le paramètre facteur d'échelle de l'élément SCALING MODE est une valeur chiffre à virgule flottante sur 32 bits, même lorsque le REAL PRECISION a retenu une virgule fixe pour d'autres réels. Dans l'ISO 8632 la précision de cette valeur à virgule flottante n'est pas indiquée lorsqu'on a choisi un nombre à virgule fixe. Sa précision doit être (0, 9, 23).
Mode sélection de couleur	--
Mode spécification largeur de ligne	--
Mode spécification largeur de marqueur	--
Mode spécification largeur de bord	--
Extension VDC	--
Couleur du fond	--

B.3.4 Eléments de commande

Précision VDC entier	16, 32
Précision VDC réel	(0, 9, 23) (1, 16, 16)
Couleur auxiliaire	--
Transparence	Transparent
Rectangle attache	--
Indicateur attache	--

B.3.5 Eléments primitives graphiques

Polyligne	La capacité minimale de prise en charge pour la longueur des listes de points est de 1024 vertex.
Polyligne disjointe	La capacité minimale de prise en charge pour la longueur des listes de pointe est de 1024 vertex.
Polymarqueur	La capacité minimale de prise en charge pour la longueur de listes de points est de 1024 vertex.
Text	La prise en charge sera assurée pour des longueurs de chaînes de 254 octets au maximum, sauf pour les enregistrements de données pour lesquels cette longueur est de 32 767 octets au maximum. Les caractères de commande de formatage ne sont pas admis dans le paramètre de chaîne.

Text restreint	La prise en charge sera assurée pour des longueurs de chaînes de 254 octets au maximum, sauf pour les enregistrements de données pour lesquels cette longueur est de 32 767 octets au maximum. Les caractères de commande de formatage ne sont pas admis dans le paramètre de chaîne.
Text joint	La prise en charge sera assurée pour des longueurs de chaînes de 254 octets au maximum, sauf pour les enregistrements de données pour lesquels cette longueur est de 32 767 octets au maximum. Les caractères de commande de formatage ne sont pas admis dans le paramètre de chaîne.
Polygone	La capacité minimale de prise en charge pour la longueur des listes de points est de 1024 vertex.
Ensemble polygone	La capacité minimale de prise en charge pour la longueur des listes de points est de 1024 vertex.
Réseau de cellules	La capacité minimale de prise en charge pour la longueur du paramètre des listes de couleurs pour l'élément CELL ARRAY est de 1 048 576. Cela permet de prendre en charge une image 1024 × 1024.
Rectangle	--
Cercle	--
Arc circulaire 3 points	--
Arc circulaire 3 points fermé	--
Centre d'arc circulaire	--
Proximité centre d'arc circulaire	--
Ellipse	--
Arc elliptique	--
Proximité d'arc elliptique	--
B.3.6 Éléments d'attribut	
Index de paquet de lignes	--
Type de ligne	Valeurs négatives interdites
Largeur de ligne	--
Couleur de ligne	--
Index de paquet de marqueurs	--
Type de marqueur	Valeurs négatives interdites
Taille de marqueur	--
Couleur de marqueur	--
Index de paquet de texte	--
Index de police du texte	Toutes les polices référencées (indexées par TEXT FONT INDEX) doivent être définies dans FONT LIST dans les paramètres de présentation des Rec. de la série T.410 du CCITT ISO 8613 ou de l'ISO 8632.
Précision du texte	--
Facteur d'expansion des caractères	--

Espacement des caractères	--
Couleur du texte	--
Hauteur des caractères	--
Orientation des caractères	--
Trajet du texte	--
Alignement du texte	--
Index du jeu de caractères	Tous les jeux de caractères référencés dans le méta-fichier (indexés par CHARACTER SET INDEX) doivent être définis dans CHARACTER SET LIST. Les seuls jeux de caractères qui peuvent être désignés dans G0 sont l'ISO 646 VIR ou les versions de l'ISO 646. Les autres caractères doivent être signés G1, G2 ou G3.
Index du jeu de caractères de remplacement	Tous les jeux de caractères référencés dans le méta-fichier (indexé par ALTERNATE CHARACTER SET INDEX) doivent être définis dans CHARACTER SET LIST.
Index du paquet de remplissage	--
Style intérieur	--
Couleur de remplissage	--
Index de hachurage	Valeurs négatives interdites.
Index de motifs	1 ... 8
Index de paquet de bords	--
Type de bord	Valeurs négatives interdites
Largeur du bord	--
Couleur du bord	--
Visibilité du bord	--
Point de référence du remplissage	--
Tableau de motifs	L'élément PATTERN TABLE a un effet non spécifié lorsqu'il apparaît dans une image consécutive à des primitives graphiques quelconques. L'élément PATTERN TABLE doit apparaître avant tout élément de primitive graphique afin que les systèmes d'interprétation sans mise à jour dynamique des motifs donne l'effet escompté. La longueur minimale du paramètre réseau couleur pour l'élément PATTERN TABLE prise en charge est 2048. Cela permet de prendre en charge 8 motifs 16 × 16; 2 motifs 32 × 32 ou 1 motif 32 × 64. Les index utilisés dans le méta-fichier doivent être définis.
Taille de motif	--
Spécification du tableau des couleurs	L'élément COLOUR TABLE a un effet non spécifié lorsqu'il apparaît dans une image consécutive à des primitives graphiques quelconques. L'élément COLOUR TABLE doit apparaître avant tout élément de primitive graphique afin que les systèmes d'interprétation sans mise à jour dynamique des couleurs donne l'effet escompté. La longueur minimale du paramètre liste de couleurs pour l'élément COLOUR TABLE prise en charge est 63. Cela permet de prendre en charge un tableau couleur à 64 entrées (0 ... 63). Tous les index utilisés dans le méta-fichier doivent être définis.
Indicateur de sources d'aspect	--

B.3.7 Eléments externes

Message	La présentation de la chaîne message peut ne pas convenir à toutes les applications. Aucune condition n'a été imposée à l'interpréteur en ce qui concerne la présentation formatée de la chaîne message. Il est seulement nécessaire de prendre en charge l'indication pas d'action. La prise en charge porte sur des longueurs maximales de 254.
Données d'application	La prise en charge sera assurée pour des chaînes de 254 octets au maximum, sauf pour les enregistrements de données qui accepteront des chaînes de 32 767 octets au maximum.

Annexe C

Bibliographie

(Cette annexe ne fait pas partie intégrante de la présente Recommandation)

- Recommandation T.50 du CCITT (1992), *Alphabet international de référence*.
- Recommandation T.51 du CCITT (1992), *Jeux de caractères codés latins pour les services de télématique*.
- Recommandation T.52 de l'UIT-T (1993), *Jeux de caractères codés non latins pour les services de télématique*.
- Recommandation T.502 du CCITT (1991), *Profil d'application de document PM-11 pour l'échange de documents contenant des caractères, sous formes retraits et formatés*.
- Recommandation T.505 du CCITT (1991), *Profil d'application de document PM-26 pour l'échange de documents à contenu mixte, sous formes retraits et formatés*.
- ISO 8571:1988, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Transfert, accès et gestion de fichiers*.
- ISO 9070:1991, *Technologies de l'information – Facilités de support SGML – Procédures d'enregistrement pour identificateurs de propriétaires de texte public*.
- ISO/CEI, TR 9573:1988, *Traitement de l'information – Facilités de support pour SGML – Techniques d'utilisation du SGML*.
- ISO 10021:1990, *Technologies de l'information – Communication de texte – Système d'échange de texte en mode message*.