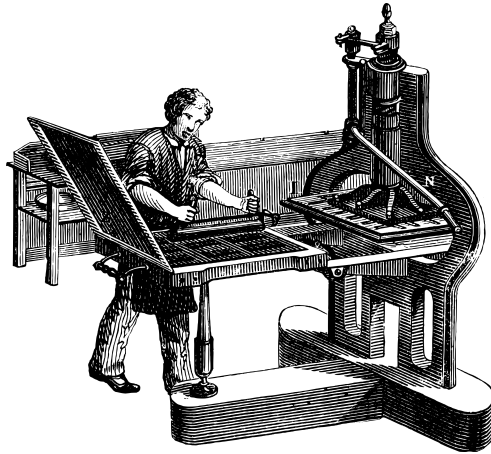


MARKUS KOHM



---

# KOMA-Script

---

TYPOGRAPHIE UNIVERSELLE AVEC  $X_{\text{T}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

---

RAYMOND ROCHEDIEU  
Juillet 2017



---

Auteurs de KOMA-Script  
Frank NEUKAM, Markus KOHM, Axel KIELHORN  
Version allemande originale datée du 13 avril 2017

Ce guide fait partie de KOMA-Script qui est libre selon les termes et conditions de licence version publique du projet L<sup>A</sup>T<sub>E</sub>X 1.3c. Le détail de cette licence est inclus dans la version originale en langue allemande du fichier `lpp1-de.txt`.

Adaptation en langue française le 13 juillet 2017  
par PAPIRAY, alias de Raymond ROCHEDIEU<sup>1</sup>.  
avec l'aimable autorisation de Markus KOHM  
en date du 19 juillet 2014.

**ISBN** : 9791092674170

**Première édition** (traduction) : octobre 2017

**Image de couverture** : gravure représentant un imprimeur sur une presse à bras Stanhope. *Grandes inventions modernes dans les sciences, l'industrie et les arts*, par Louis Figuier, troisième édition (1874). Gravure : Guiguet, Pégard.

---

1. Pour me contacter en français : [ecrire@papieray.fr](mailto:ecrire@papieray.fr), ou : [raymond@rochedi.eu](mailto:raymond@rochedi.eu).



# Table des matières

<b>KOMA-Script adapté <math>\LaTeX</math></b>	<b>xv</b>
Préalable de Papiray . . . . .	xv
Conditions d'utilisation . . . . .	xvii
Limitation de responsabilité . . . . .	xvii
Version KOMA-Script . . . . .	xvii
À propos de cet ouvrage . . . . .	xviii
KOMA-Script3 . . . . .	xix
Remerciements . . . . .	xx
Remarques sur la version 3.22a . . . . .	xxi
<b>Introduction</b>	<b>xxv</b>
Avant-propos . . . . .	xxv
Structure de documents . . . . .	xxvi
L'histoire de KOMA-Script . . . . .	xxviii
Outils d'installation . . . . .	xxix
Messages d'erreur, réponses aux questions . . . . .	xxix
<b>Méthode d'installation</b>	<b>xxxiii</b>
Installation à partir d'archives . . . . .	xxxiii
Installation d'un arbre <code>texmf</code> décompressé . . . . .	xxxiv
Installation d'une distribution minimale . . . . .	xxxiv
<b>1 KOMA-Script pour auteurs</b>	<b>1</b>
1.1 Recalculer avec l'extension <code>typearea.sty</code> . . . . .	2
1.2 Principes de base de la mise en page . . . . .	2
1.3 Mise en page par division . . . . .	7
1.4 Mise en page en dessinant un cercle . . . . .	8
1.5 Option de sélection précoce ou tardive . . . . .	9

1.6	Compatibilité avec les versions précédentes . . . . .	11
1.7	Réglage de <code>typearea</code> et mise en page . . . . .	12
1.8	Choix et réglage de la taille de papier . . . . .	29
1.9	Conseils . . . . .	32
<b>2</b>	<b>Les principales classes</b>	<b>37</b>
2.1	Option de sélection précoce ou tardive . . . . .	38
2.2	Compatibilité avec les versions précédentes . . . . .	40
2.3	Mode création . . . . .	41
2.4	Mise en page . . . . .	41
2.5	Choix de la taille de police du document . . . . .	42
2.6	Description de textes . . . . .	43
2.7	Titre du document . . . . .	49
2.8	Le résumé . . . . .	57
2.9	Table des matières . . . . .	58
2.10	Mise en page des paragraphes . . . . .	63
2.11	Reconnaissance des pages paire et impaire . . . . .	66
2.12	Styles prédéfinis d'entête et pied de page . . . . .	67
2.13	Pages vierges . . . . .	73
2.14	Notes de pied de page . . . . .	76
2.15	Délimitations . . . . .	82
2.16	Structure du document . . . . .	84
2.17	Citation et dicton . . . . .	107
2.18	Listes . . . . .	109
2.19	Mathématiques . . . . .	120
2.20	Environnement flottant pour figure et table . . . . .	121
2.21	Note de bordure . . . . .	142
2.22	Annexe . . . . .	143
2.23	Bibliographie . . . . .	144
2.24	Index . . . . .	148
<b>3</b>	<b>La lettre <code>scrlltr2</code></b>	<b>151</b>
3.1	Les variables . . . . .	151
3.2	Les pseudo-longueurs . . . . .	156
3.3	Option de sélection précoce ou tardive . . . . .	157
3.4	Compatibilité avec les versions antérieures . . . . .	159
3.5	Mode de conception . . . . .	160

3.6	Mise en page . . . . .	161
3.7	Structure générale des documents de format lettre . . . . .	162
3.8	Choix de la taille de police pour la lettre . . . . .	173
3.9	Feuille de papier à lettre . . . . .	177
3.10	Mise en page du paragraphe . . . . .	212
3.11	Détection des pages paires et impaires . . . . .	213
3.12	Entête et pied de page avec style prédéfini . . . . .	214
3.13	Pages intercalaires . . . . .	219
3.14	Notes de pied de page . . . . .	221
3.15	Listes . . . . .	225
3.16	Mathématiques . . . . .	228
3.17	Environnements flottants de tableaux et figures . . . . .	229
3.18	Notes de marge . . . . .	229
3.19	Salutations finales . . . . .	230
3.20	Fichiers en option de la classe Lettre . . . . .	232
3.21	Adresse de fichiers et lettres circulaires . . . . .	239
<b>4</b>	<b>Entête et pied de page sclayer-scrpage</b>	<b>245</b>
4.1	Option de sélection précoce ou tardive . . . . .	246
4.2	Hauteurs de l'entête et du pied de page . . . . .	247
4.3	Personnaliser le texte . . . . .	248
4.4	Utilisation de pages avec styles prédéfinis . . . . .	251
4.5	Manipulation des styles de page prédéfinis . . . . .	261
<b>5</b>	<b>Le jour de la semaine avec scrdate</b>	<b>275</b>
<b>6</b>	<b>L'heure actuelle avec scrtime</b>	<b>281</b>
<b>7</b>	<b>Accès aux fichiers adresses scraddr</b>	<b>283</b>
7.1	Vue d'ensemble . . . . .	283
7.2	Utilisation . . . . .	284
7.3	Paq avec options d'avertissement . . . . .	286
<b>8</b>	<b>Création de fichiers d'adresses</b>	<b>287</b>
<b>9</b>	<b>Développer les entités scrextend</b>	<b>289</b>
9.1	Option de sélection précoce ou tardive . . . . .	290
9.2	Compatibilité avec les versions précédentes . . . . .	291

9.3	Des fonctions étendues en option . . . . .	293
9.4	Développer en mode création . . . . .	293
9.5	Sélection de la taille de police du document . . . . .	293
9.6	Étiquetage du texte . . . . .	294
9.7	Titres de documents . . . . .	296
9.8	Détection des pages paires et impaires . . . . .	302
9.9	Choix d'un style de page pré-défini . . . . .	303
9.10	Pages intercalaires vierges . . . . .	303
9.11	Notes de pied de page . . . . .	306
9.12	Citations et maximes . . . . .	311
9.13	Listes . . . . .	312
9.14	Notes de marge . . . . .	314
<b>10</b>	<b>Soutien à la pratique du droit</b>	<b>315</b>
10.1	Préalable . . . . .	315
10.1.1	Généralités sur <code>scrjura</code> . . . . .	315
10.1.2	Contenu . . . . .	316
10.1.3	Remarques juridiques . . . . .	316
10.2	Le paq <code>scrjura</code> version 2014 . . . . .	316
10.2.1	Aperçu . . . . .	317
10.2.2	Options pour charger le paq . . . . .	318
10.2.3	Options de titre de colonne . . . . .	319
10.2.4	Le paragraphe . . . . .	320
10.2.5	Références . . . . .	321
10.2.6	Alinéas et numérotation de la composition . . . . .	324
10.2.7	Personnaliser <code>scrjura</code> avec Kile et sous Lyx . . . . .	324
10.2.8	Environnements contractuels supplémentaires . . . . .	325
10.2.9	Et encore ... . . . .	326
10.2.10	Problèmes connus . . . . .	327
10.2.11	Mise en œuvre . . . . .	327
10.3	Généralités sur <code>scrjura</code> version 2016 . . . . .	356
10.3.1	Choix d'option précoce ou tardive . . . . .	356
10.3.2	Répertoires . . . . .	356
10.3.3	Environnement pour les contrats . . . . .	357
10.3.4	Section juridique . . . . .	358
10.3.5	Propriétés de <code>dummy</code> . . . . .	359
10.3.6	Alinéas . . . . .	361



10.3.7 Archives . . . . .	363
10.3.8 Références croisées . . . . .	364
10.3.9 Environnements de contrats supplémentaires . . . . .	366
10.3.10 Prise en charge linguistique . . . . .	369
10.3.11 Un exemple détaillé . . . . .	370
10.4 Développement . . . . .	375
<b>11 KOMA-Script pour utilisateurs avancés et experts</b>	<b>377</b>
11.1 Fonctions de base de <code>scrbase</code> . . . . .	377
11.2 Charger le <code>paq</code> . . . . .	378
11.3 Caractéristiques des familles et extensions . . . . .	379
11.4 Exécution conditionnelle de <code>scrbase</code> . . . . .	393
11.5 Identificateur de la langue . . . . .	397
11.6 Identification de KOMA-Script . . . . .	399
11.7 Extensions du noyau $\text{\LaTeX}$ . . . . .	401
11.8 Compétences mathématiques de $\varepsilon\text{-TeX}$ . . . . .	401
<b>12 Dépendances des paqs avec <code>scrfile</code></b>	<b>403</b>
12.1 À propos des dépendances des paqs . . . . .	403
12.2 Actions avant et après le chargement . . . . .	404
12.3 Remplacer les fichiers lors de la lecture . . . . .	410
12.4 Empêcher le chargement de fichier . . . . .	413
<b>13 Remplacer et enregistrer les fichiers avec <code>scrwfile</code></b>	<b>417</b>
13.1 Modification fondamentale du noyau $\text{\LaTeX}$ . . . . .	418
13.2 Le système de fichiers . . . . .	419
13.3 Le clonage des entrées avec <code>CloneFileWrite</code> . . . . .	419
13.4 Note sur le niveau de développement . . . . .	422
13.5 Options connues de <code>paq</code> incompatibles . . . . .	422
<b>14 Gérer les répertoires en utilisant <code>tocbasic</code></b>	<b>423</b>
14.1 Commandes de base . . . . .	423
14.2 Génération d'une table des matières ou d'un répertoire . . . . .	428
14.3 Configuration des entrées de répertoire . . . . .	437
14.4 Commandes internes pour auteurs . . . . .	449
14.5 Un exemple complet . . . . .	452
14.6 Tout dans une seule instruction . . . . .	455

<b>15 Améliorer les « paqs » étrangers avec scrhack</b>	<b>463</b>
15.1 Note sur le développement . . . . .	463
15.2 Options de sélection précoce ou tardive . . . . .	464
15.3 Utilisation de tocbasic . . . . .	465
15.4 Faux espoirs avec \@ptsize . . . . .	466
15.5 Cas particulier avec hyperref . . . . .	467
15.6 Traitements incompatibles de \textwidth et \textheight . .	467
15.7 Cas spécial de nomencl . . . . .	468
<b>16 Définition des niveaux et styles de page avec sclayer</b>	<b>469</b>
16.1 Note sur le niveau de développement . . . . .	470
16.2 Choix de l’option de sélection précoce ou tardive . . . . .	470
16.3 Informations élémentaires . . . . .	472
16.4 Déclaration de niveaux . . . . .	474
16.5 Déclaration et gestion des styles de page . . . . .	486
16.6 Hauteurs de l’entête et du pied de page . . . . .	498
16.7 Influence des styles de page définis . . . . .	499
16.8 Interfaces pour l’utilisateur final . . . . .	506
<b>17 Des options supplémentaires du paq sclayer-scrpage</b>	<b>509</b>
17.1 Influencer les styles de page définis . . . . .	509
17.2 Définir vos propres paires de style de page . . . . .	514
17.3 Définir de simples styles de page avec entête et pied de page en trois parties . . . . .	516
17.4 Définition des styles de page complexes . . . . .	519
<b>18 Notice sur sclayer-notecolumn</b>	<b>523</b>
18.1 Information pour le développement . . . . .	524
18.2 Choix de l’option de sélection précoce ou tardive . . . . .	525
18.3 Textes et marques . . . . .	525
18.4 Déclarer de nouvelles colonnes de notes . . . . .	526
18.5 Créer un mémo . . . . .	531
18.6 Édition forcée des colonnes de notes . . . . .	535
18.7 Extrait du manuel sclayer-notecolumn . . . . .	536
<b>19 Informations complémentaires sur typearea.sty</b>	<b>543</b>
19.1 Caractéristiques expérimentales . . . . .	543

19.2	Commandes pour experts . . . . .	545
19.3	Fichiers typearea.cfg de paramètres locaux . . . . .	546
19.4	Options plus ou moins obsolètes . . . . .	546
<b>20</b>	<b>Informations supplémentaires sur les classes</b>	<b>547</b>
20.1	Compléments d'information à propos du mode d'emploi . .	547
20.2	Interaction avec d'autres paqs . . . . .	548
20.3	Instructions pour experts . . . . .	548
20.4	Options et instructions obsolètes . . . . .	571
<b>21</b>	<b>Informations supplémentaires</b>	<b>573</b>
21.1	Les pseudo-longueurs . . . . .	573
21.1.1	Marques de pliage . . . . .	578
21.1.2	Entête de lettre . . . . .	581
21.1.3	Adresse . . . . .	582
21.1.4	Les extensions de l'expéditeur . . . . .	584
21.1.5	Occupation de ligne . . . . .	585
21.1.6	Sujet de la lettre . . . . .	586
21.1.7	Salutations finales . . . . .	587
21.1.8	Pied de lettre . . . . .	588
21.2	Variables pour utilisateur avancé . . . . .	589
21.3	Les différences dans les styles de scrletter . . . . .	590
21.4	Le traitement des fichiers avec lco-scrletter . . . . .	592
21.5	Fichiers .ico pour utilisateur avancé . . . . .	592
21.5.1	Suivi de la taille du papier . . . . .	592
21.5.2	Visualisation de positions . . . . .	593
21.6	Prise en charge linguistique . . . . .	596
21.7	De scrlettr obsolète à l'actuel scrlettr2 . . . . .	599
<b>22</b>	<b>Création de répertoires avec splitidx et SplitIndex</b>	<b>601</b>
22.1	Résumé . . . . .	601
22.2	Introduction . . . . .	602
22.3	Le programme SplitIndex . . . . .	603
22.3.1	Objectif . . . . .	603
22.3.2	Mise en œuvre . . . . .	603
22.4	Utilisation du paq splitidx . . . . .	604
22.4.1	Installation . . . . .	604

22.4.2	Marquage des entrées d'index . . . . .	606
22.4.3	Suppression de la génération de répertoires multiples	606
22.4.4	Personnalisation des entrées de répertoires . . . . .	607
22.4.5	Commandes automatiques de répertoire personnalisé	608
22.4.6	Prévenir l'expansion prématurée des entrées . . . . .	608
22.4.7	Inclure les répertoires générés dans votre document	609
22.4.8	Paramètres type des répertoires générés . . . . .	610
22.4.9	Exemples . . . . .	611
22.4.10	Fractionnement des fichiers répertoires . . . . .	614
22.4.11	Utilisation de splitindex.pl . . . . .	616
22.4.12	Utilisation de splitindex.jar . . . . .	619
22.4.13	Utilisation de splitindex.tex . . . . .	619
22.4.14	Fusionner les index . . . . .	620
22.5	Mise en œuvre de splitidx . . . . .	621
22.5.1	Options . . . . .	621
22.5.2	Définition d'une entrée d'index . . . . .	622
22.5.3	Édition d'un ou de plusieurs index . . . . .	626
	<b>Références</b>	<b>635</b>
	<b>Bibliographie</b>	<b>639</b>
	<b>Liste des figures</b>	<b>643</b>
	<b>Liste des tables</b>	<b>647</b>
	<b>Index</b>	<b>651</b>





# KOMA-Script adapté L<sup>A</sup>T<sub>E</sub>X

## Préalable de Papiray

L<sup>A</sup>T<sub>E</sub>X a été créé par des Américains pour des Américains. Il permet d'écrire du texte, de mettre en page facilement des équations et, entre autres, de faire des présentations diapos... Pour l'utiliser convenablement il faut charger des paqs qui permettent de l'adapter à notre langue, par exemple `babel` et son option `french`. Mais la mise en page des formats de feuilles de papier et surtout la typographie sont très différentes et L<sup>A</sup>T<sub>E</sub>X n'est pas vraiment adapté à notre format A4.

KOMA-Script est un outil d'écriture fonctionnant avec L<sup>A</sup>T<sub>E</sub>X<sup>1</sup>, créé pour la langue et la typographie allemandes par Markus KOHM, conçu pour la mise en page de texte n'intégrant pas de formule mathématique<sup>2</sup>, intégrant un ensemble de classes et de paqs. Au fil de l'évolution du projet, les classes sont employées, depuis 1994, par des utilisateurs d'autres langues.

Je ne suis ni germaniste émérite, ni même angliciste, ce document n'est donc pas une traduction, *stricto sensu*, mais une interprétation compilée de la version originelle et de sa variante en langue anglaise. Plusieurs auteurs ont participé à la migration vers l'anglais. Certains d'entre eux sont de langue maternelle anglaise et d'autres, comme l'inventeur de cet ouvrage, Markus KOHM, de langue allemande.

---

1. Je ne différencie pas volontairement les dérivés de T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt... et autres, existants ou à venir.

2. Argument repris par Maïeul ROUQUETTE, dans le bandeau de l'ouvrage édité en 2012 chez ATRAMENTA : *L<sup>A</sup>T<sub>E</sub>X appliqué aux sciences humaines, le seul livre sur L<sup>A</sup>T<sub>E</sub>X sans une seule équation*, (ce qui, déjà à l'époque, n'est pas tout à fait vrai!).

Lorsque l'on me rappelle que plus de 270 millions de personnes dans le monde s'expriment dans le même langage que MOLIERE, je n'hésite plus et c'est en pensant à nos gens<sup>3</sup> que je me lance, en espérant que ce travail sera amélioré par des locuteurs ou des experts indigènes qui seront les bienvenus en tout temps. J'ai par exemple utilisé *package* à la place des mots paquet (*Paket*) ou paquetage, pour désigner l'ensemble cohérent d'un programme constitué pour exécuter une fonction définie nécessaire à l'exécution d'une classe. Toutes réflexions faites, plutôt que l'anglicisme *pack*, je retiens l'apocope *paq* qui me semble plus judicieuse. De plus l'homophonie du terme avec son synonyme (très rare) me comble. J'utilise également le mot *empagement* qui désigne soit la totalité de la surface imprimée à l'intérieur des marges (la page plus le titre courant et la pagination) soit, et c'est selon, dans une mise en page de multiples opérations, telle la détermination des marges.

Alors ? Positionner le texte ou déterminer les marges ?

S'il est vrai que fixer l'un conditionne inévitablement les autres, je ne balance pas sur la bonne interprétation de cette définition et décide donc de conserver *empagement* pour désigner l'espace englobant texte, entête<sup>4</sup> et pied de page, de créer, encore par apocope, le substantif *empage* pour désigner la mise en page c'est-à-dire l'encombrement du texte seul, appelé aussi, par les professionnels de la typographie, la zone et de garder le mot *page* pour désigner sans ambiguïté<sup>5</sup> le feuillet.

En définitive, attaché à respecter l'esprit initial de l'ouvrage, j'ai aussi et surtout vérifié que les formules enveloppées d'accolades, de crochets, de parenthèses, d'espaces et autres, sont retranscrites avec exactitude. Je me suis investi avec cœur dans cette tâche, ce qui ne préjuge en rien des contresens, lapsus, erreurs, et coquilles que j'ai pu commettre, à l'insu de mon plein gré<sup>6</sup>.

Merci de ne pas m'en tenir rigueur.

PAPIRAY

---

3. Personnes appartenant à un même groupe, ici la communauté francophone.

4. Mot masculin issu de la nouvelle orthographe.

5. Nouvelle orthographe (orthographe rectifiée de 1990).

6. Rappelez-vous de l'émission « Les Guignols de l'info », en 1998, et du coureur cycliste Richard VIRENQUE, *dopé à l'insu de son plein gré* lors du Tour de France.



## Conditions d'utilisation

KOMA-Script est sous licence publique L<sup>A</sup>T<sub>E</sub>X projet. Une traduction en langue anglaise fait partie du paq (fichier `lpp1.txt`). L'original (en langue allemande) est également disponible (fichier `LPPL-de.txt`). Il est valable pour tous les pays de langue germanique, alors que dans tous les autres pays, c'est le texte `lpp1.txt` qui est appliqué.

Ce document et le paq KOMA-Script sont fournis « tels quels », sans garantie d'aucune sorte.

## Limitation de responsabilité

Il n'existe aucune garantie d'utilisation du logiciel dont les propriétés sont décrites dans cet ouvrage. Les auteurs et l'interpréteur de ce guide n'assument aucune responsabilité pour les dommages accidentels liés à, ou découlant de l'utilisation des informations ou programmes figurant ici. La création du texte et les reproductions d'images ont été faites avec beaucoup de soins.

Néanmoins, des erreurs ou omissions ne peuvent pas être totalement exclues.

Les marques déposées et les noms de produits cités dans ce manuel sont propriété de leurs détenteurs respectifs, y compris les dessins, images et photos.

Toutes suggestions d'amélioration seront les bienvenues, l'auteur vous en est reconnaissant à l'avance.

## Version KOMA-Script

Ce guide fait partie de KOMA-Script, qui est libre selon les termes et conditions de licence version publique du projet L<sup>A</sup>T<sub>E</sub>X 1.3c. Le détail de cette licence est inclus dans le fichier KOMA-Script `LPPL-de.txt`, version originale en langue allemande. La distribution de ce manuel est permise à

la condition de conserver la version complète de **KOMA-Script**, une version partielle nécessite l'autorisation des auteurs.

Il existe une traduction anglaise réalisée par Kevin PFEIFFER ainsi que, Gernot HASSENPFUG, Krickette MURABAYASHI, Markus KOHM, Jens-Uwe MORAWSKI, Harald BONGARTZ, Georg GRANDKE, Raimund KOHL, Stephan HENNIG, Karl-Heinz ZIMMER, Christoph BIER, Jana SCHUBERT, Jens HÜHNE et Melvin HENDRIX.

Une mise à jour des instructions de mise en page ainsi qu'une édition augmentée des commandes de **KOMA-Script** sont disponibles, en langue allemande, aux Éditions DANTE e.V. et LEHMANN Media (voir [KM12]).

## À propos de cet ouvrage

L'adaptation et la mise en forme de l'ouvrage en langue française est l'œuvre de PAPIRAY, alias de Raymond ROCHEDIEU, avec l'autorisation de Markus KOHM en date du 19 juillet 2014.

Ce travail est réalisé sur la base de la version allemande du 07 février 2015, modifiée le 16 avril, le 15 septembre, le 3 octobre de la même année puis le 14 juin 2016 et le 02 janvier 2017 pour aboutir à la version 3.23 du 13 avril 2017 (la version anglaise est également datée du 13 avril 2017), dans le cadre du projet  $\LaTeX$  sous licence :

$\LaTeX$  Project Public License version 1.3c.

La version française est soumise à nos amis de Framasoft. Que tous, ici, soient remerciés pour leurs conseils avisés et notamment Mireille, Frédéric (Fred), Gatien (Asta), Jean-Bernard (Goofy), Martin (Bromind), Yann et Christophe qui se sont investis dans cette tâche en ne m'épargnant aucune proposition de corrections, parfois discutables mais finalement toujours bien-fondées.

## **KOMA-Script3**

KOMA-Script3 diffère, à bien des égards, des versions de KOMA-Script de 2.5 à 2.98. Les éditions précédentes vous sont déjà familières ? alors ce manuel est fait pour vous, pas seulement en raison des fonctions inédites, mais aussi pour leur approche originale dans cette nouvelle édition.

Au cours de ces dernières années, une majorité de lecteurs qui semblaient dans l'ensemble assez satisfaits, m'ont fait des commentaires sur certaines instructions paraissant trop courtes et demandant à être plus développées, ce que j'ai rendu possible en divisant certains chapitres. Ainsi, la classe `lettre` contenait, paraît-il, trop peu d'exemples. J'ai pris en compte cette remarque et révisé complètement le chapitre avec ajout de nombreuses illustrations.

D'autres critiques portaient sur l'obligation de consulter le manuel pour les principales catégories d'utilisation de cette même classe, remarque dont j'ai tenu compte dans cette nouvelle version : chaque chapitre ne traite que d'une question dont le sujet n'est plus référencé dans d'autres chapitres, comme précédemment. Grâce à la poursuite du développement d'exemples plus détaillés, le lecteur peut mieux voir les différences entre les préférences paramétrables et les options de configuration par défaut. Les nombreuses illustrations permettent de définir avec exactitude ce que l'on veut obtenir au final et d'adapter les paramètres à partir du code source associé. Pour les grands exemples, images ou tables qui nécessitent plus d'informations, il peut être nécessaire de suivre un renvoi à un autre chapitre.

La principale observation concernait la division des chapitres, bien que les sujets traités au commencement d'un document soient également cités au début du chapitre, sans affecter les options de classe spécifiées. L'auteur de nombreuses options, que je suis, avait oublié que ces mêmes options sont définies dans l'entête du document. J'ai donc décidé d'organiser les chapitres entièrement par thèmes, ce qui conduit à une hiérarchie plus plate dans le plan.

Outre ces modifications perceptibles par les lecteurs ayant déjà étudié les éditions précédentes : le nouveau livre apparaît nettement plus épais. Cette augmentation, justifiée par la mise en œuvre de vos observations,

amis lecteurs, est due également aux nombreuses et nouvelles fonctions possibles de `KOMA-Script3`.

Une critique positive des publications précédentes a comme conséquence l'écriture, toujours avec rigueur, des connaissances de base dans un langage facilement accessible, malgré l'aridité du sujet traité.

La version numérique peut comporter quelques limitations, par exemple des informations considérées comme essentielles pour utilisateurs avancés, ou aptes à faire d'un utilisateur lambda un utilisateur avancé, réservées à la version écrite du livre. Cela signifie qu'il est possible (mais improbable) qu'un lien indique une page du manuel non reproduite ici<sup>7</sup>.

## Remerciements

L'auteur de `KOMA-Script` exprime ses remerciements à Frank NEUKAM, sans qui la famille `KOMA-Script` n'aurait jamais existé.

Je suis redevable aux nombreuses personnes qui ont contribué, et avec leur permission, je voudrais mentionner spécifiquement Torsten KRÜGER qui a été un des meilleurs bêta-testeurs que j'ai jamais eu, Elke, Jana, Ben et Edoardo pour leur travail qui a notamment amélioré la convivialité de `scrllttr2` et `scrpage2`. Un grand merci à tous ceux qui m'ont encouragé, permettant la correction d'erreurs ou la mise en œuvre de fonctionnalités supplémentaires.

Je remercie spécialement les fondateurs et membres de DANTE (association germanophone d'utilisateurs T<sub>E</sub>X ) coupables de la propagation de T<sub>E</sub>X et L<sup>A</sup>T<sub>E</sub>X, ainsi que des paqs `KOMA-Script` ; de même les contributeurs actifs sur la liste de diffusion T<sub>E</sub>X-DL (voir [Wik]) du groupe de discussion `Usenet.de` et les nombreux forums de L<sup>A</sup>T<sub>E</sub>X sur l'internet qui apportent les réponses aux questions posées sur `KOMA-Script`.

Enfin, je remercie tous ceux qui ont participé à la création de ce livre, avec une mention particulière adressée à Jens-Uwe MORAWSKI, mon co-auteur, aux bêta-testeurs et relecteurs. En outre, je remercie DANTE e.V.

---

7. PAPIRAY précise : j'ai fait « avec », mon document source parait entier et je l'ai complété d'exemples libres attribués à Markus KOHM, issus d'annexes glanées au hasard du net. Je n'ai aucun accès à l'édition originale de DANTE E.V.

et LEHMANN Media, à savoir Klaus HOEPPNER et Christoph KAEDER qui ont rendu possible le fait que personne n'ait plus à imprimer ces écrits, grâce à [KM12], version de ce manuel disponible dans une forme liée de près de 200 pages.

Markus KOHM, NECKARHAUSEN - Septembre 2008.

## **Remarques sur la version 3.22a**

La publication du guide KOMA-Script3, après une révision complète de sa structure et de son contenu, a provoqué ces dernières années, quelques commentaires qui me sont parvenus.

Manifestement cela valait la peine d'élargir le chapitre sur les lettres à d'innombrables exemples ou encore de rendre le contenu des autres classes largement indépendant.

Bien que ces diverses sections, souvent avec des exemples modifiés ou omis, figurent à plusieurs reprises dans les instructions, cette procédure a donc été appliquée de manière plus cohérente à partir de KOMA-Script v.3.22 et transférée à d'autres chapitres.

Le manuel KOMA-Script gratuit présente, au-delà d'une nouvelle image, un nombre importants de liens nouveaux.

Hormis la description immédiate, un clic sur une macro KOMA-Script conduit maintenant, par exemple, à une explication appropriée respective. Les références croisées avec le numéro de la section concernée ou le numéro de page sont néanmoins toujours présentes. Cela est dû au livre imprimé et sert également pour l'orientation. Cependant, les lecteurs de cette version libre doivent supporter certaines restrictions, ainsi, quelques informations considérées comme principales pour les utilisateurs avancés ou capables de faire d'un utilisateur un utilisateur avancé sont réservées à la version allemande du livre.

Cela signifie seulement que dans ce guide les liens qui conduisent à une page sont tout simplement mentionnés mais inopérants.

De plus, la version gratuite est limitée à l'expression appropriée, c'est-à-dire que l'accent est plutôt mis sur l'utilisation du parallèle avec l'écran pour travailler sur un document.

En outre elle n'a pas de mise en page optimisée, mais est comme un premier projet, dans lequel à la fois le paragraphe et le saut de page sont assez pauvres dans certains cas. Les optimisations correspondantes font l'objet de l'édition imprimée [Koh14a].

En attendant je trouve pour le manuel des remarques assez peu critiques et même pour les classes et les paqs, il n'y a guère de demandes de nouvelles opportunités.

Pour moi, cela signifie que les souhaits de connaissances que l'utilisateur exprime depuis plusieurs années stagnent.

J'ai donc principalement mis en œuvre ce que je suppose être utile. Les commentaires que je reçois sur ces nouvelles opportunités se limitent principalement à la critique que les **hacks** anciens, qui se fondent sur les caractéristiques non documentées du programme, ne fonctionneront pas dans certains cas.

Cependant, le plaisir de constater que des solutions d'urgence peu nettes ne sont plus nécessaires était à peine exprimé. Par conséquent, j'ai décidé de limiter les extensions et les améliorations à **KOMA-Script** de plus en plus aux choses qui sont explicitement demandées par les utilisateurs.

Je me pose une question : en 25 ans, **KOMA-Script** aurait-il simplement atteint un niveau dans lequel il répond à toutes les exigences ?

Malheureusement, le développement rétrograde avec messages d'erreur n'est pas forcément agréable et il est fréquent d'observer qu'un découvreur de problème ne me le signale pas immédiatement mais aborde le sujet dans certains forums Internet où il trouve souvent des contournements plus ou moins intelligents au problème posé.

La performance est très satisfaisante, mais conduit généralement au fait que le problème ne m'est jamais rapporté et donc ne sera jamais vraiment éliminé. Le fait que ces mesures palliatives peuvent elles-mêmes devenir éventuellement un problème avéré a déjà été mentionné dans le paragraphe précédent.

Sporadiquement on peut se retrouver isolé s'il n'existait heureusement une tierce manière me rapportant ces questions et qui affecte très peu les contributions individuelles juste dans les forums.

Le contact direct avec celui auquel le problème est survenu, n'est souvent pas possible, bien qu'il serait en partie souhaitable.

Par conséquent, il est explicitement redemandé de m'indiquer les bugs supposés directement et uniquement en allemand ou en anglais. Ici, la perfection linguistique est moins importante.

Le message doit être raisonnablement compréhensible et le problème facilement identifiable.

En bref et dans la règle, un exemple de code doit être, indépendamment de la langue, intégré de fait. Pour moi un contact direct et des requêtes sont possibles en cas de besoin.

S'il vous plaît ne comptez pas sur quelqu'un qui aurait déjà signalé certains points du problème que l'on peut supposer résolu lorsque vous l'avez vous-même déclaré.

Pour plus de détails sur les entreprises des messages d'erreur, voir le premier chapitre du manuel.

Markus KOHM, NECKARHAUSEN - Janvier 2017.  
à des températures glaciales





# Introduction

Ce chapitre contient des informations importantes sur `KOMA-Script`, son histoire qui commence des années avant la première version, sa structure et son usage.

## Avant-propos

`KOMA-Script` est un ensemble complexe développé en langue allemande, basé sur `TEX`, soit à partir d'une classe, soit à partir de paq. Il est conçu d'une manière homologue aux classes `standard` de `LATEX` (voir chapitre 3), avec des moyens particuliers, commandes, environnements et paramètres des classes, mais ses fonctionnalités vont bien au-delà de la capacité des classes `standard`. Certains de ces éléments sont considérés complémentaires aux fonctions de base du noyau de `LATEX`.

On a pu constater que la documentation précédente de `KOMA-Script`, aussi vaste soit-elle, était insuffisante et imparfaite.

Et puis, `KOMA-Script` n'est pas enseigné : pas de professeur, pas d'élèves, pas de leçons, pas de sélection ni de réglage du matériel d'enseignement. Il serait sans doute facile d'écrire de la documentation pour un groupe cible. Mais la difficulté serait de créer autant de guides nécessaires que de groupes cibles identifiés, un guide approprié pour les scientifiques comme pour le secrétariat de la poissonnerie, même si cela paraît être vraiment une tâche impossible, avec, comme résultat, beaucoup de compromis. Vous avez été nombreux à m'interpeller, j'ai pris en compte vos réclamations pour améliorer l'outil en lui ajoutant de nouvelles instructions.

## Structure de documents

Ce manuel est divisé en plusieurs parties : pour les utilisateurs, pour les utilisateurs et les experts avancés et une annexe avec des informations plus détaillées et des exemples pour ceux qui veulent en savoir plus.

La première partie s'adresse à tous. Cela signifie que vous trouverez là des informations pour les nouveaux usagers de  $\LaTeX$ . En particulier, cette partie est enrichie avec de nombreux exemples destinés à l'utilisateur de base pour illustrer les explications. Ne craignez pas de tester ces exemples par vous-même pour trouver comment fonctionne **KOMA-Script**. Cependant ces instructions ne sont pas une introduction à  $\LaTeX$  et les novices sont donc encouragés à consulter les documents tels que [DGS+12]. Pour ceux qui se remettent à  $\LaTeX$  depuis l'époque de  $\LaTeX$  2.09, il est recommandé au minimum de lire [tea05b]. Réintégrer une ancienne version de  $\LaTeX$  2.09 est au moins recommandé en étudiant un autre livre sur  $\LaTeX$ . Des recommandations de littérature se trouvent par exemple dans [Wik] dont le champ d'application est considérable. Il est donc souhaitable de lire le document au moins une fois et de consulter les problèmes antécédents.

La seconde partie est destinée aux utilisateurs avancés de **KOMA-Script**, ceux qui sont déjà familiers avec  $\LaTeX$ , ou qui ont passé du temps à travailler avec **KOMA-Script** et qui veulent maintenant comprendre un peu mieux comment l'outil fonctionne, comment il interagit avec d'autres paqs et la façon possible de résoudre des problèmes spécifiques, en complément des descriptions des catégories de la partie I prises selon certains aspects et expliquées en détail. À cette fin, la documentation conçue pour les usagers pointus et les experts est complétée par celle des paqs qui sont normalement réservés et cachés, ceux qui font leur travail systématique et discret sous la surface des classes et des paqs d'utilisateur, spécifiquement prévus pour une utilisation par d'autres auteurs de classes et de paqs.

L'annexe, qui figure uniquement dans l'édition allemande du livre<sup>1</sup>, est destinée aux utilisateurs à qui tout le reste ne suffit pas. Des rudiments de typographie s'y trouvent développés pour les aider à prendre leurs propres décisions. Il existe en outre, pour les aspirants auteurs de paqs, des exemples non pas destinés à être simplement copiés, mais aussi à

---

1. Cette annexe ne figure ni dans l'édition anglaise, ni dans le présent ouvrage.

fournir des instructions de base  $\text{\LaTeX}$ , ainsi que des connaissances utiles à la planification et à la mise en œuvre de projets.

La division du manuel en chapitres vous aidera à lire, en priorité, les pièces qui ont pour vous de l'intérêt. Pour ce faire, les informations sur les classes et les paqs spécifiques ne sont pas réparties mais concentrées, pour chaque cas, dans une **section** avec des renvois faisant références à d'autres éléments du projet. Si l'utilisation d'une seule classe ou d'un seul paq **KOMA-Script** vous intéresse, vous pouvez probablement éviter la lecture de la totalité du guide. Les trois classes principales **scrbook**, **scrreprt** et **scrartcl**), qui coïncident en grande partie, sont introduites et traitées ensemble dans une **section** commune, le chapitre 2.

Dans la version originale de cet ouvrage, leurs différences sont clairement mises en évidence, le cas échéant, par une note appelée dans la marge, note à laquelle j'ai renoncé dans cette adaptation car chaque classe et son paq ont généralement leur propre chapitre. L'inconvénient de cette approche est qu'il faut supporter des retours dans certains chapitres sur ce qui est déjà connu, pour découvrir **KOMA-Script** dans sa totalité.

Ce qui peut être l'opportunité d'approfondir vos connaissances.

Dans la version originale, différentes polices sont utilisées pour mettre en évidence des choses distinctes. J'ai choisi d'écrire le nom des paqs et des fichiers avec `\texttt` et d'utiliser le paq `listing` pour identifier clairement les commandes<sup>2</sup>.

Les instructions des environnements variables et pseudo-longueurs sont mises en évidence uniformément. L'espace réservé pour chaque paramètre est présenté comme une valeur concrète du paramètre.

Par exemple, un environnement est lancé par `\begin{environnement}` alors que `\begin{document}` spécifie comment l'environnement commence concrètement le document. Une valeur spécifique est ensuite donnée comme paramètre de la commande `\begin`.

La documentation initiale de **KOMA-Script** est en allemand, traduite en français pour votre commodité, mais les commandes, les instructions d'environnements, les variables, les options et pseudo-longueurs sont en anglais, comme il est d'usage dans le monde de  $\text{\LaTeX}$ .

---

2. PAPIRAY précise : j'ai adopté la même présentation dans le présent ouvrage.

## L'histoire de KOMA-Script

Au début des années 1990, Frank NEUKAM cherche pour une conférence une méthode de publication de notes d'un instructeur. À cette époque,  $\LaTeX$  était encore en version 2.09, il n'y avait pas de distinction de classes ni de paqs, et tous les styles étaient en langue anglaise. Les « standards » de documents ne lui apparurent pas optimaux et n'offraient pas toutes les commandes et les environnements dont il avait besoin. Dans le même temps, Frank traite également des questions de typographie, en particulier avec [Tsc87] et il était clair, pour lui, qu'il fallait créer un outil pour exécuter n'importe quel style de document, mais aussi une famille de style qui suit les règles de la typographie européenne. **Script** était né.

L'auteur de **KOMA-Script** a trouvé le ton juste sur son sujet en 1992/93. Contrairement à Frank NEUKAM, il a souvent traité des documents en format A5. À cette époque, les modèles de ce format n'étaient pas soutenus, même par défaut. Par conséquent, il ne fallut pas longtemps pour qu'il fasse les premiers changements dans le script. Ils figurent également dans le nouveau **Script2** publié en décembre 1993 par Frank.

Dès la mi-1994,  $\LaTeX$  devient disponible avec de nombreuses évolutions. Son utilisateur est confronté à un choix : limiter l'utilisation à un mode compatible avec  $\LaTeX$ , ou renoncer à tout script.

Cette situation conduit Markus à mettre sur pied un nouveau paq  $\LaTeX$  publié le 7 Juillet 1994, **KOMA-Script**, que Frank déclare officiellement successeur de **Script**, quelques mois plus tard. Je ne m'attarderai ni sur la confusion qui existait à la succession, ni sur le choix de ce nouveau nom. Il faut mentionner que, à l'origine, **KOMA-Script** a été publié sans la classe lettre, lacune corrigée en décembre 1994 par Axel KIELHORN. Plus tard encore, Axel SOMMERFELDT a créé le premier **scrguide** réel de **KOMA-Script**, en langue allemande, suivi par un guide en langue anglaise que l'on doit à Werner LEMBERG.

Depuis, le temps a passé.  $\LaTeX$  n'a changé que de manière mineure, contrairement à son environnement. **KOMA-Script** s'est développé bien au-delà de ce que l'on pouvait imaginer à sa création. Aujourd'hui, il n'est plus réservé aux utilisateurs de langue allemande, mais utilisé partout en Europe, Amérique du Nord, Australie et Asie. Ses usagers sont non

seulement à la recherche d'un outil offrant un résultat typographiquement adapté et agréable, mais ils apprécient également son nouveau point fort : soutenir une flexibilité qui a conduit à l'intégration, tant bien que mal, de nouvelles et nombreuses macros dans la documentation existante. Il était donc temps d'éditer un manuel entièrement révisé, à l'origine de la troisième version du guide.

## **Outils d'installation**

La gestion des paqs `KOMA-Script` fonctionne parfaitement sous les trois distributions `TEX` les plus importantes, `MacTEX`, `MiKTEX` et `TEX Live`. Il est recommandé de faire l'installation et les mises à jour de `KOMA-Script` en utilisant ces outils. Néanmoins, l'installation manuelle sans utiliser le gestionnaire de paq est décrite dans le fichier `INSTALL.txt`, qui fait partie de chaque distribution `KOMA-Script`. Il serait bon, également, de lire la documentation fournie avec la distribution `TEX` que vous utilisez.

Depuis quelques temps, on peut aussi trouver dans [KDP] des versions intermédiaires de `KOMA-Script` pour les principales distributions. Pour leur installation, il est indispensable de respecter les instructions locales.

## **Messages d'erreur, réponses aux questions**

Si vous croyez que vous avez trouvé une erreur dans ce modèle, dans l'une des classes, dans l'un des paqs ou une autre partie de `KOMA-Script`, vous devez procéder comme suit. Tout d'abord, contrôler sur `CTAN` pour voir si une version plus récente de `KOMA-Script` est disponible. Le cas échéant, installer cette nouvelle version et vérifier si l'erreur ou le problème est toujours présent.

S'il ne s'agit pas d'un bogue dans la documentation et que l'erreur ou le problème persiste après la mise à jour, créer, comme il est indiqué dans [Wik], un exemple minimal. Procéder comme décrit ci-dessous. Souvent, il s'agit d'un problème limite qui peut être résolu par l'utilisateur, que ce soit une erreur d'application ou non.

Il est possible d'identifier le paq ou la classe concrètement à l'origine de la difficulté et de voir s'il s'agit réellement d'un problème `KOMA-Script`, ce qui peut faire l'objet d'une vérification supplémentaire par un test : changer la classe `KOMA-Script` par une classe `standard` correspondante. Après cela, il est nécessaire d'envoyer le rapport de bogue à l'auteur de `KOMA-Script` ou à l'auteur du paq en cause. Vous devez aussi étudier, une fois de plus et soigneusement, les instructions sur le paq approprié, les classes et les constituants `KOMA-Script`, ainsi que les conseils [Wik] car une solution à votre problème existe peut-être déjà.

Vous avez trouvé une erreur inconnue, ou d'autres choses de base que vous estimez utiles ou nécessaires, contacter l'auteur de `KOMA-Script`, mais n'oubliez pas les conseils suivants :

- le problème se produit aussi quand une classe `standard` est utilisée à la place d'une classe `KOMA-Script` ? Dans ce cas, l'origine de l'erreur n'est probablement pas dans `KOMA-Script` et il paraît judicieux de poser la question dans un forum public, une liste de diffusion ou *usenet*.
- quelle version `KOMA-Script` utilisez-vous ? L'information pertinente peut être trouvée dans le fichier journal que  $\LaTeX$  réalise à chaque document d'une classe `KOMA-Script` utilisée.
- quel système d'exploitation et quelle distribution  $\TeX$  sont utilisés ? Ces informations semblent superflues dans un paq indépendant du système comme `KOMA-Script` ou  $\LaTeX$ , mais peuvent certainement jouer un rôle important.
- quel est exactement le problème ou l'erreur ? Ne soyez pas concis et n'hésitez pas à en préciser les détails. Il est souvent utile d'en expliquer les origines.
- qu'est-ce qu'un exemple minimal complet ? Tout le monde peut en créer un facilement, étape par étape, en désactivant un à un tous les contenus et paqs superflus du document et en commentant chaque opération. À la fin, il reste un document qui charge uniquement les paqs concernés et comprendra seulement les parties nécessaires à l'examen de l'exemple. Les illustrations doivent être chargées et les instructions correspondantes remplacées. Supprimer les parties en commentaire avant l'émission, ajouter la première ligne `\rule` des fichiers de la liste de commande et accomplissez un passage de  $\LaTeX$

qui permet d'obtenir un fichier journal contenant un résumé des paqs utilisés et qui ajoute la description du fichier journal et de l'exemple minimal complet.

- n'envoyez ni les paqs ni les fichiers DVI,PDF ou PS. Si la description d'erreur, y compris le fichier journal minimal et l'exemple dépassent quelques dizaines de kilo-octets, c'est que, selon toutes probabilités, vous vous êtes trompé quelque part. Sinon, envoyer votre message à `komascript@gmx.info`.
- il vous arrivera de poser une question sur **KOMA-Script** ou en public, par exemple, sur `de.comp.text.tex` ou sur le forum pour [PDK]. Même dans ce cas, vous devez respecter les points ci-dessus, à l'exception de l'inclusion du fichier journal. Au lieu de cela, ajouter la liste et les versions des paqs à partir du fichier journal.
- en cas de message d'erreur, citer également le fichier journal.

Noter que les paramètres non parfaits typographiquement ne constituent pas des erreurs. Pour des raisons de compatibilité, les réglages subsistent, si possible, dans les nouvelles versions **KOMA-Script**.

En outre, la typographie est aussi une question de langue et de culture. Les Préférences de **KOMA-Script** fournissent donc inévitablement, et par défaut, un compromis.

Une fois habitués, vous souhaiterez peut-être exécuter des tâches plus difficiles, qui vont au-delà de l'enseignement des connaissances de base et ne sont donc pas incluses dans ces instructions. Cependant, sur le site du projet de documentation **KOMA-Script** [KDP], vous trouverez d'autres exemples conçus pour les utilisateurs avancés de  $\text{\LaTeX}$  mais moins adaptés pour les débutants.





# Méthode d'installation

Il existe diverses manières de distribuer KOMA-Script. La distribution principale, provenant des mainteneurs de KOMA-Script, consiste en un arbre `texmf` complet (connu aussi sous l'appellation de TDS-Baum), fourni sous forme d'archives ZIP. Il peut également être distribué dans d'autres formats. Quand le texte suivant se réfère à une archive, il ne se réfère pas donc nécessairement à une archive ZIP. En créant une nouvelle archive, les distributeurs sont fortement encouragés à préserver la structure et le contenu de l'arbre `texmf` pour que ces instructions restent valables. Et nous leur demandons de fournir leurs propres instructions d'installation.

La distribution de l'arbre `texmf` décompressé étendu est aussi permise. S'il devait en résulter une modification de la structure ou du contenu de l'arbre `texmf` (dans la limite permise par la licence) rendant partiellement ou complètement inapplicables les présentes instructions d'installation, le distributeur devra fournir de nouvelles instructions.

L'utilisateur doit donc vérifier si le distributeur a inclus des nouvelles instructions d'installation. En toute logique, leur nom devrait commencer par `installD`, immédiatement suivi par le nom du distributeur ou un acronyme. Un dossier `readme` devrait de même avoir été créé.

## Installation à partir d'archives

Si vous avez reçu l'archive de KOMA-Script avec un arbre `texmf` complet, l'installation est tout à fait facile. Vérifier que l'archive contient vraiment l'arbre complet et jeter un coup d'œil à son contenu qui doit se composer

de trois répertoires : `docLATEX KOMA-Script`, `TEX LATEX KOMA-Script` et source `LATEX KOMA-Script`, le dernier avec des sous-répertoires.

Dans chacun de ces répertoires, il y a plusieurs dossiers. Il devrait au moins y avoir, avec le chemin `doc\LATEX\KOMA-Script\readme`, un fichier `readme` qui peut être, le cas échéant, extérieur à l'archive.

Quand vous déballez une telle archive, garder la structure répertoire de l'arbre mais sans créer de nouveau sous-répertoire, que ce soit dans l'arbre `texmf` local ou dans un arbre `texmf` personnalisé. Des informations sur l'arbre `texmf` local et la création d'un arbre personnalisé peuvent être trouvées dans les instructions de votre système `TEX`.

Enfin, rafraîchissez la base de données des noms de fichiers directement avec `texhash` ou `mktextlsr`. Le programme `texconfig` est utilisé par les systèmes `XYLATEX` et basés sur `XYLATEX`. Avec `MiKTEX` vous trouverez la mise à jour dans le menu `\mikOptions` de l'application. Selon les permissions définies pour votre répertoire d'installation, vous devrez le faire avec les droits administrateur ou `root`. Pour plus d'informations, voir le manuel de votre système `TEX`.

## Installation d'un arbre `texmf` décompressé

Si `KOMA-Script` est présent comme arbre `textmf` étendu, vous pouvez l'enregistrer dans le système comme un arbre `texmf` supplémentaire ou en copier le contenu en préservant la structure de répertoire dans un arbre `texmf` privé ou local.

Là encore, il est nécessaire de rafraîchir la base de données de noms de fichiers. Voir le dernier paragraphe de la section précédente pour plus d'informations.

## Installation d'une distribution minimale

L'installation d'une distribution minimale incluant seulement ces fichiers strictement nécessaires énumérés dans le fichier `manifest.txt` n'est pas aussi simple. Pour ceci, `Gnu-make` et d'autres outils `UNIX` sont essentiels.

En gardant cela à l'esprit, le processus est le suivant :

- créer un fichier `Makefile.cfg`,
- copier une des lignes qui commence par `#installtexmf` du fichier `Makefile.baseinit` dans ce nouveau fichier,
- enlever le caractère dièse `#` qui débute la ligne,
- éditer cette ligne pour que le nom du répertoire où l'arbre `texmf` doit être installé apparaisse après le signe égal.

AVERTISSEMENT : Aucun oblic (`/`) avant ou controblic<sup>1</sup> (`\`) à la fin du nom de répertoire.

- exécuter votre programme `make`,
- après l'exécution de `make` avec succès et sans erreurs, exécuter `make install`.

Finalement, la mise à jour ou le rafraîchissement des noms de fichiers de la base de données sont faits, avec `XYTEX` et les systèmes basés sur `XYTEX`, par le programme `texconfig` ou par `texhash` ou encore par `mktextlsr`. Le bouton correspondant à la mise à jour se trouve dans les options de l'application `MiKYTEX`. Selon les permissions qui ont été enregistrées dans votre répertoire d'installation, vous agirez comme administrateur ou à la racine. Pour plus d'informations, voir le manuel de votre système `TEX`.

- exécuter la commande `TEX scrmain.ins`. Répondre aux questions posées avec « y » pour « oui ».
- copier tous les fichiers nouvellement créés dont les noms finissent en `.cls`, `.sty` ou `.ico` dans le répertoire `\TEX\LaTeX\KOMA-Script`.
- effectuer une mise à jour finale comme indiqué précédemment.

Dernier recours : installation à partir d'une distribution minimale.

Si vous n'avez pas l'outil `gnu-make` ou les autres outils UNIX nécessaires à la méthode d'installation utilisant `make` il est possible d'installer le paq `KOMA-Script` manuellement à partir de la distribution minimale. Il vous faudra seulement faire une plus grande partie du travail vous-même.

Créer tout d'abord les nouveaux répertoires de l'arborescence `texmf` dans lesquels seront installés `.source`, `.doc` et `.tex \LaTeX\KOMA-Script`.

---

1. Mot valise masculin invariable, un controblic, que j'utilise pour remplacer, dans la nouvelle orthographe, le terme anglais `antislash`. Par conséquent, j'écris également, au masculin, un oblic pour désigner, en français, le `slash` anglais.

- copier les fichiers de la distribution KOMA-Script dans le répertoire source `\LaTeX\KOMA-Script` nouvellement créé,
  - copier tous les fichiers avec l'extension `.pdf`, `.html` ou `.txt` et le fichier `readme` dans le répertoire `doc\LaTeX\KOMA-Script`,
  - aller dans le répertoire source `\LaTeX\KOMA-Script`,
  - exécuter la commande `tex scrmain.ins`. Répondre aux questions avec « y » pour « oui ».
  - copier tous les fichiers nouvellement créés avec l'extension `.cls`, `.sty` ou `.ico` dans le répertoire `\TeX \LaTeX \KOMA-Script`,
  - enfin, effectuer une actualisation du fichier base de données. Dans `TeX` et les systèmes basés sur `TeX`, cela se fait avec le programme `texconfig` ou directement avec `texhash` ou `mktexlsr`. Avec `MiKTeX`, le bouton se trouve dans l'application `\Miktex-Options.pdf`.
- Selon le répertoire d'installation, la mise à jour n'est possible qu'en tant que `root` ou en tant qu'administrateur. Pour plus d'informations, se référer au manuel de votre système `TeX`.

---

Texte allemand de Markus KOHM, 2006

Texte anglais de Kevin PFEIFFER et Markus KOHM, 2006

Interprétation française par Raymond ROCHEDIEU dit PAPIRAY, 2017

---





# Chapitre 1

## KOMA-Script pour auteurs

Dans cette partie, se trouvent des informations pour auteurs d'articles, de rapports et de livres. Il est supposé qu'un utilisateur moyen est peu intéressé par le fonctionnement de KOMA-Script, ni par le niveau où se situent les difficultés de programmation. En outre pour cet utilisateur, les options obsolètes et les instructions qui en découlent ont peu d'intérêt. Il préférera sans doute savoir comment réaliser un travail et aura, peut-être, un intérêt pour la typographie.

Les passages contenant informations et justifications supplémentaires, qui pourraient donc avoir moins d'intérêt pour le lecteur impatient, sont dans l'ouvrage original en police sans serif. Si vous êtes intéressés par plus d'informations sur la mise en œuvre, les effets secondaires avec d'autres paqs, les options et instructions obsolètes, référez vous à la partie II du guide qui décrit les fonctions créées pour les auteurs de paqs et de classes. Je n'ai écarté aucune `section`, même si les chapitres 3 (la lettre `scrlettr2`), 5 (jour et heure avec `scrdate`), 6 (`scrttime`), 7 (accès aux fichiers d'adresses avec `scraddr`) et 8 (création de fichiers d'adresses à partir d'une base de données d'adresses) ne semblent pas, à mes yeux, d'un intérêt évident pour l'usage que j'aimerais faire de KOMA-Script. Toutefois, et malgré ce que je viens d'écrire, je n'ai pas repris le « court » chapitre qui traite de la mise en page et des formats d'enveloppes destinés au Japon.<sup>1</sup>

---

1. Que les utilisateurs concernés par ce sujet m'en excusent et se réfèrent aux documents originaux en allemand ou en anglais.

## 1.1 Recalculer avec l'extension `typearea.sty`

De nombreuses classes de L<sup>A</sup>T<sub>E</sub>X fournissent à l'utilisateur une division des espaces et de la zone de texte, y compris les classes `standard`. Dans les classes `standard` de la distribution d'origine, les limites dépendent de la taille de la police choisie. En outre, il existe des paqs comme `geometry` (voir [Ume00]), qui donnent à l'utilisateur un contrôle total permettant de paramétrer la zone de texte et les bordures.

KOMA-Script est livré avec le paq `typearea` qui fonctionne quelque peu différemment. Les paramètres et les automatismes proposés, basés sur un concept typographique, font qu'il est plus facile pour l'utilisateur de faire un bon choix.

Il convient de noter que `typearea` repose sur `scrbase`, paq commenté dans la partie II, chapitre 11, mais que la plupart des instructions dans ce document sont plutôt destinées aux auteurs de classes et de paqs.

## 1.2 Principes de base de la mise en page

En regardant une seule page d'un livre ou d'un ouvrage imprimé, on remarque, à première vue : un entête<sup>2</sup>, un corps avec des bordures et un pied de page. On peut ajouter les espaces entre l'entête et le corps et entre le corps et le pied. Dans le langage technique des typographes et suivant le type de typographie, le corps (que j'ai appelé *empage* dans le préalable de cet ouvrage) prend le nom de zone ou zone de saisie. La répartition de ces zones (*empage*, corps, ainsi que leur inter-relation) est décrite comme la mise en page ou composition, voir la figure 1.1 page 4. Je n'y connais pas grand chose en typographie, mais ça ne me semble pas très technique comme définition<sup>3</sup>.

Dans la littérature, plusieurs algorithmes et heuristiques sont proposés et discutés [Koh02] pour construire une bonne mise en page.

---

2. PAPIRAY précise : merci la nouvelle orthographe (orthographe rectifiée de 1990) et si vous l'avez déjà oublié, je rappelle que cet « entête » est du genre masculin.

3. Cette précision est de Markus KOHM.



Souvent, on trouve une méthode qui fonctionne avec des diagonales et leurs intersections. Le résultat recherché est que le rapport d'aspect du type de zone correspond au rapport d'aspect de la page. Dans un document recto, les marges gauche et droite doivent avoir la même largeur, alors que les marges supérieure et inférieure doivent être dans le rapport de 1/2. Dans un document à double page, tel un livre, il est bon de noter que la totalité de la largeur de la marge centrale doit être la même que celle de chacune des deux marges extérieures. Chacune des pages contribue donc à la moitié de la marge centrale.

Dans le paragraphe précédent, la page mentionnée est souvent prise à tort pour le format qui équivaldrait à la taille du papier. Quand on regarde un document relié, on peut voir qu'une partie du papier disparaît dans la reliure et que cette partie de la feuille n'est plus visible. Cependant, pour la zone, le format du papier n'a pas d'importance, c'est plutôt celui de la page visible qui en a. Il est donc clair que la partie cachée par la reliure sur le format de papier doit être ajoutée à la bordure intérieure. Nous l'appelons correction de reliure, elle fait partie du calcul de la gouttière, sans bord intérieur visible.

La correction de reliure dépend du processus de production et ne peut pas être déterminée en général. Il s'agit d'un paramètre redéfini pour chaque processus de production. Dans le domaine professionnel, la valeur ne joue qu'un rôle mineur, car l'impression est généralement faite sur des feuilles de papier grand format qui seront réduites en conséquence. La coupe est faite de manière que les relations ci-dessus pour la page **recto-verso** soient maintenues.

Il n'y a marge de gauche et marge de droite que dans l'impression recto. En conséquence, bord interne et bord externe n'existent qu'en impression **recto-verso**. Dans les équations, nous travaillons avec la moitié de la marge intérieure car la marge intérieure complète est un élément de la double page complète.

Nous connaissons maintenant les relations entre les différentes parties de la composition dont nous ignorons encore la largeur et la hauteur. Nous pouvons choisir une de ces dimensions et, avec le format du papier et la taille de la page, en déduire toutes les autres par la résolution de plusieurs équations mathématiques.

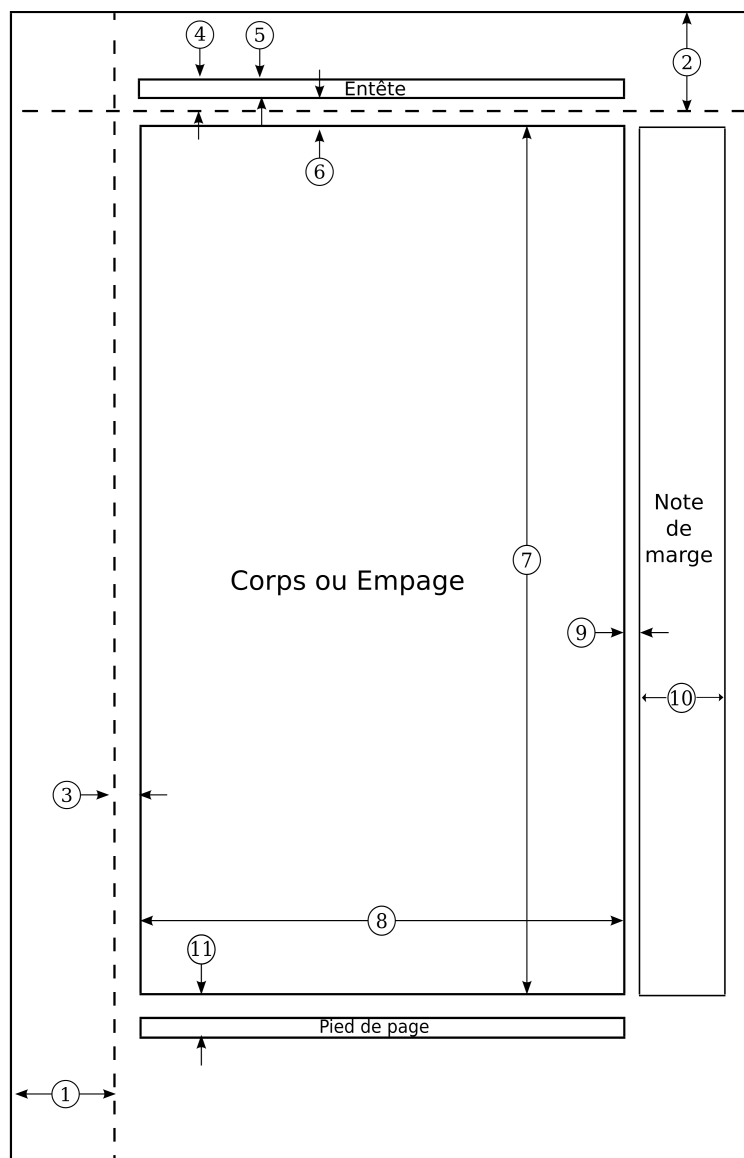


FIGURE 1.1 – Empagement, marges et empage

- 1. hoffset; 2. voffset; 3. oddsidemargin; 4. topmargin;
- 5. headheight; 6. headsep; 7. textheight; 8. textwidth;
- 9. marginparset; 10. marginparwidth marginparpush;
- 11. footskip

La question de la largeur de la zone de saisie est également discutée dans la littérature. La largeur optimale dépend de divers facteurs :

- taille, largeur et type de la police utilisée,
- utilisation d'un premier plan,
- longueur des mots,
- espace disponible.

TABLE 1.1 – Ratios de principes utilisés pour la mise en page

ratio hauteur / largeur	= ratio hauteur / largeur de page
marge supérieure / hauteur page	= 1 : 2
marge gauche / marge de droite	= 1 : 1
demi bord interne / bord externe	= 1 : 2
largeur de l'empagement	= largeur du papier - reliure
correction sup. + marge inf.	= hauteur de page - zone
marge gauche + marge de droite	= largeur de page - largeur de zone
marge intérieure + bord extérieur	= largeur de page - largeur de zone
marge int.+ correction reliure	= gouttière

L'influence de la police devient claire quand vous pensez *empatement*. Il consiste en de petits traits horizontaux aux extrémités des barres de lettres. Les lettres finissant verticalement à la base de la ligne de texte ont tendance à entraîner le regard hors de cette ligne. Les empattements horizontaux aident à maintenir le regard aligné sur la ligne de base et facilitent le flot de lecture, mais surtout à retrouver rapidement le début de la ligne après le saut depuis la fin de ligne précédente.

Ainsi une ligne de texte identique peut être plus longue dans le cas d'une police avec empatement que dans celui d'une police sans empatement.

L'espacement signifie la distance verticale entre les lignes de texte. Dans  $\text{\LaTeX}$  l'espacement standard correspond à 20 % de la taille de la police. Avec des commandes comme `\linespread` ou mieux à l'aide d'un paq comme `setspace` (voir [FT11]), ces rapports peuvent être modifiés. Un espacement de ligne plus large permet à l'œil de mieux suivre cette ligne. Un écart très large facilite la lecture, mais perturbe l'œil qui doit parcourir

de longues distances entre les lignes. En outre, un effet de bande très clair est perturbant et désagréable pour le lecteur car la valeur de gris de la page est, dans ce cas, altérée. Malgré tout, un espacement important permet une longueur de ligne plus importante.

Vous cherchez des valeurs concrètes pour des bonnes longueurs de ligne et avez peut-être trouvé des informations différentes, en fonction de l'auteur, qui peuvent être bien fondées dans sa langue maternelle. L'œil saute de mot en mot habituellement et les mots courts rendent cette tâche plus facile. Tous langages et polices confondus, une longueur de ligne de 60 à 70 caractères, espaces et ponctuation comprises, constitue un bon compromis. Cela suppose un espacement de ligne bien choisi. Avec les réglages par défaut de  $\LaTeX$  vous n'avez, normalement, pas de soucis à vous faire. Une longueur plus importante est réservée à l'usage de bibliophages aguerris, habitués à de longues heures quotidiennes de lecture.

Mais même alors, des lignes de 80 caractères ne sont pas raisonnables, et l'espacement doit être adapté avec 5 à 10 % supplémentaires comme orientation. Pour certaines polices telle Palatino qui nécessite déjà une longueur de ligne normale avec 5 % d'espacement supplémentaire, des valeurs supérieures sont nécessaires.

Avant d'examiner la mise en page de la construction proprement dite, il y a encore des petites choses que vous devez savoir. Dans  $\LaTeX$ , la première ligne de la zone de texte d'une page ne commence pas au niveau du bord supérieur du bloc de texte. Le niveau de référence de la ligne est défini avec une distance minimale du bord supérieur de la zone de texte.

Spécificités des commandes  $\LaTeX$  `\raggedbottom` et `\flushbottom` :

- la première, `\raggedbottom`, indique que la dernière ligne d'une page doit être positionnée là où cela a été calculé, ce qui peut provoquer une variation de la position verticale de la dernière ligne d'une page à l'autre, variation qui peut atteindre la hauteur d'une ligne en cas de cumul d'ajustement de titres, de figures, de tableaux ou similaires, ce qui est généralement indésirable dans les conditions d'impression double-face,
- la deuxième, `\flushbottom`, dit que la dernière ligne est toujours au niveau du bord inférieur de la zone de texte et que, pour atteindre ces compensations verticales,  $\LaTeX$  peut avoir à s'étendre, ce qui

est possible en additionnant les distances verticales extensibles. Une telle distance peut être, par exemple, l'espacement des paragraphes, mémorisée lors de l'utilisation de paragraphes sans espacements qui est la seule mesure verticale expansible, la hauteur de la zone de texte doit être un multiple de la hauteur de la ligne de texte, augmentée de la distance de la première ligne à partir du bord supérieur de la zone de texte.

En ce qui concerne les bases, les deux sections suivantes présentent les constructions possibles permises par KOMA-Script.

## 1.3 Mise en page par division

La manière la plus facile de réaliser une zone de texte dans le même rapport que la page est la suivante :

- évaluer la partie BCOR nécessaire et diviser verticalement le reste de la page en rangées DIV d'égale hauteur,
- diviser la page horizontalement dans le même nombre de DIV en bandes de même largeur,
- prendre la ligne supérieure de la partie supérieure et les deux rangées du bas sur le bord inférieur. Pour une impression recto-verso, ne pas oublier la correction de reliure à l'intérieur et les deux colonnes sur les bords extérieurs,
- BCOR est ensuite ajouté à la marge intérieure.

Il reste à déterminer la zone de texte dans la page. La largeur ou la hauteur des bordures fixe automatiquement le résultat de la zone de texte à partir du nombre de colonnes DIV. Comme les marges ont toujours besoin de trois lignes, DIV doit nécessairement être supérieur à trois.

Le texte occupe horizontalement et verticalement au moins deux fois autant d'espace que les bords, DIV pourrait même être inférieur à 9. Avec cette valeur, la construction permet la mise en page classique dite au neuvième (voir la figure 1.2 page suivante).

La mise en page est réalisée avec le paq `typearea` pré-régulé pour le format papier A4. Selon la taille de police, des valeurs différentes pour DIV peuvent être prises, voir la table 1.1 page 5. En l'absence de correction, on applique

1	2	3	4	5	6	7	8	9		9	8	7	6	5	4	3	2	1
2																		2
3																		3
4																		4
5																		5
6																		6
7																		7
8																		8
9																		9

FIGURE 1.2 – Construction classique d’une grille avec application de DIV par neuf et après déduction d’une correction de liaison.

BCOR=0 pt et les résultats correspondent à peu près aux valeurs de mise en page de la table 1.

En plus des valeurs prédéfinies, vous pouvez indiquer BCOR et DIV en chargeant le paq spécifié en option (voir section 2.6). Une commande vous permet de calculer un ensemble explicitement et de passer ces valeurs comme paramètres existants. Le paq `typearea` offre automatiquement la possibilité de déterminer la valeur optimale de DIV qui dépendra de la taille de la police fixée au moment du recalcul.

## 1.4 Mise en page en dessinant un cercle

Outre la méthode de construction décrite précédemment, il existe dans la littérature une méthode un peu plus classique dite « méthode médiévale » ou « du Moyen Âge ». Dans cette méthode, vous trouverez les mêmes valeurs, non seulement dans la forme du rapport d’aspect, mais il est supposé, en outre, que l’optimum est obtenu lorsque la hauteur de la zone de texte est égale à la largeur de la page. La méthode exacte peut être

lue dans [Tsc87]. Un inconvénient de ce procédé montre que la largeur de la zone de texte n'est plus dépendante de la police de caractères. Il est donc judicieux pour l'auteur ou le compositeur de sélectionner une bonne police pour cette zone de texte.

Avec `typearea` cette construction est modifiée dans la mesure où par la sélection d'une excellente, mais généralement absurde, valeur de `DIV` ou d'une option spéciale de `paq`, une valeur de `DIV` est déterminée avec laquelle la mise en page qui en résulte correspond aussi précisément que possible, aux valeurs parfaites de la page du « livre médiéval ». On a donc, encore une fois, recours à la méthode de construction par division.

## 1.5 Option de sélection précoce ou tardive

Dans cette section, une particularité de `KOMA-Script` est présentée, qui concerne non seulement `typearea` mais aussi les autres `paqs` et classes : l'utilisateur peut trouver toutes les informations sur un `paq` ou une classe, et cette section se retrouve, presque à l'identique, dans plusieurs chapitres. Les utilisateurs qui ne sont pas intéressés par ces instructions, mais qui souhaitent juste avoir une vue globale de `KOMA-Script`, peuvent éviter cette étude plus approfondie du manuel.

```
\documentclass[liste d'options]{KOMA-Script-classe}  
\usepackage[liste d'options]{paq-liste}
```

En  $\text{\LaTeX}$ , il est prévu que les utilisateurs spécifient les options de classe sous la forme d'une simple liste de mots-clés séparés par des virgules comme un argument optionnel de `\documentclass`. Ces préférences sont passées à tous les `paqs` qui comprennent l'option. De même, les utilisateurs spécifient les `paqs` sous la forme d'une simple liste des mots-clés séparés par des virgules comme argument optionnel de `\usepackage`. `KOMA-Script` étend à d'autres possibilités le mécanisme d'options pour les classes et les `paqs`. En outre, la plupart de ces options ont une valeur qui n'est pas de forme nécessairement mais peut avoir la forme `option=valeur`. Sauf qu'il existe une différence de traitement data `\documentclass` et `\usepackage` `KOMA-Script` comme dans [Tea05b] ou toute introduction de  $\text{\LaTeX}$ , par exemple [DGS+12].

Lorsque vous utilisez une classe KOMA-Script, il n'est pas nécessaire d'en spécifier les options pour éviter une charge superflue. Cela est dû au fait que la classe charge, par défaut, les paqs qui sont déjà installés.

Le réglage des options avec `\documentclass` ou `\usepackage` révèle un inconvénient majeur, les déclarations de longueurs, de compteurs et autres doivent être suivies strictement pour ne pas générer un message d'erreur. En effet, une extension de L<sup>A</sup>T<sub>E</sub>X dans la valeur d'une option est utilisée par ces instructions avant même que la valeur de KOMA-Script soit adoptée. L'affectation de valeurs avec des extensions ou compteurs de L<sup>A</sup>T<sub>E</sub>X ne devrait jamais se faire via `\documentclass` ou `\usepackage` mais toujours à l'aide des commandes `\KOMAoptions` ou `\KOMAoption`.

<pre>\KOMAoptions{liste d'options} \KOMAoption{option}{liste de valeurs}</pre>
--

KOMA-Script offre aussi la possibilité de changer la valeur des options avec la plupart des classes et des paqs. Vous pouvez modifier les valeurs sélectivement avec la déclaration `\KOMAoptions`. Chaque option de la liste des options a la forme `option=valeur`.

Certaines options ont aussi une valeur par défaut (`defaultValue`) qui sera, en cas d'oubli, appliquée automatiquement. Certaines options peuvent en avoir simultanément plusieurs. Pour ces options il est possible d'assigner à l'aide de `\KOMAoption` un ensemble de valeurs. Dans la liste des valeurs, les valeurs individuelles seront séparées les unes des autres par des virgules.

*Si une valeur contient le signe égal ou une virgule, elle doit être incluse entre des accolades.*

KOMA-Script utilise les commandes `\FamilyOptions` et `\FamilyOption` de la famille KOMA pour la réalisation de cette possibilité. Voir partie II, article 11.3. Mettre une option avec une valeur non valide produira une erreur. Si L<sup>A</sup>T<sub>E</sub>X est exécuté dans un mode interactif, il s'arrêtera à cet endroit. La saisie de `h` (`help`) affiche un écran d'aide qui donne les valeurs valides pour l'option correspondante.



## 1.6 Compatibilité avec les versions précédentes

Réaliser ses documents dans le code source présente un intérêt lié au fait que les futurs développements de  $\LaTeX$  devraient donner exactement le même résultat. Néanmoins, dans certains cas l'amélioration des paqs ou la correction de bugs peut entraîner des modifications du comportement, en particulier pendant la transition vers la nouvelle version, ce qui n'est, en général, ni voulu, ni souhaitable.

```
version=value
version=first
version=last
```

Depuis la `version3.01b` c'est à `typearea` de choisir si votre code source doit être intégré à la version courante de  $\LaTeX$  ou figurer dans une nouvelle version et, si vous souhaitez, participer aux améliorations de cette nouvelle version, vous pouvez choisir la version de `KOMA-Script` compatible avec la version de l'option, soit avec la version soutenue la plus ancienne, soit avec la `version=2.9`, `2.9t` ou une version ultérieure. Si vous spécifiez une version inconnue comme valeur, un avertissement sera émis et pour des raisons de sécurité, `version=first` sera conservée.

La valeur `version=last` sélectionne la version la plus récente à chaque exécution de  $\LaTeX$ , mais cette utilisation peut poser des problèmes de compatibilité avec l'évolution future de  $\LaTeX$ . L'option sans valeur a la même fonction. C'est aussi la valeur par défaut tant que vous n'utilisez pas d'option obsolète. Dans ce cas, `KOMA-Script` passe automatiquement en `version=first` et un avertissement émis explique comment éviter cette commutation. Autrement, vous pouvez sélectionner un réglage différent compatible avec l'option obsolète.

La question de la compatibilité affecte principalement des questions de création. De nouvelles fonctionnalités non liées à la transition, sont aussi disponibles si vous avez sélectionné la compatibilité d'option de version plus ancienne. Elle n'a pas d'effet sur les modifications par l'élimination de bugs lorsque vous utilisez une version plus récente.

Cependant, si une compatibilité s'avère indispensable, pour cause de bug, il est préférable d'enregistrer et d'archiver la version de `KOMA-Script` utilisée avec votre document. Il est important de noter que la version

après le chargement du paq `typearea` ne peut pas être modifiée. L'accès à `\KOMAOPTIONS` ou `\KOMAOPTION` n'est pas disponible.

## 1.7 Réglage de `typearea` et mise en page

Le paq de `typearea` propose deux interfaces utilisateur différentes qui influent sur la construction. Le réglage le plus important consiste à en spécifier les options qui peuvent ainsi être définies de différentes manières. Dans cette section, la classe `protokol` sera utilisée. Ce n'est pas une classe KOMA-Script, mais une classe hypothétique. Ce guide est basé sur le cas idéal, avec une classe correspondante disponible pour chaque tâche.

`BCOR=correction`

Avec l'aide de l'option `BCOR=correction` vous entrez la valeur absolue de la correction de reliure, à savoir la largeur de la zone utilisée pour cette reliure. Cette valeur est prise en compte automatiquement dans la méthode de construction et ajoutée à la marge intérieure ou gauche, selon le montage choisi. Vous pouvez la corriger en lui spécifiant une unité de format reconnu par  $\TeX$ .

Exemple : vous créez un rapport financier assemblé avec une reliure type agrafe portefeuille de 7,5 mm, imprimé au format A4. La taille du papier ne sera pas suffisante en ajoutant la reliure. Ainsi, il vous faudra écrire :

```
\documentclass[A4paper]{report}
\usepackage[BCOR=7.5mm]{typearea}
```

ou

```
\documentclass[A4paper, BCOR=7.5mm]{report}
\usepackage{typearea}
```

pour spécifier `BCOR` comme option globale.

*Lorsque vous utilisez une classe de KOMA-Script, `typearea` est chargé automatiquement et il n'est pas nécessaire de l'appeler.*

```
\documentclass[BCOR=7.5mm]{scrbook}
```

*L'option `A4paper` peut être omise parce qu'elle est la valeur par défaut pour toutes les classes KOMA-Script.*

Si vous choisissez une nouvelle valeur, celle qui est utilisée, par exemple :

```
\documentclass{scrbook}
\KOMAOPTIONS{BCOR=7.5mm}
```

et si la classe est déjà en cours de chargement, ce sont les paramètres par défaut de `scrbook` qui sont utilisés. Lorsque, avec une des commandes `\KOMAOPTIONS` ou `\KOMAOPTION`, vous changez de paramètres, une mise en page nouvelle est calculée automatiquement en utilisant les paramètres de marge nouveaux. Lorsque vous utilisez l'une des classes de `KOMA-Script`, assurez-vous que cette option est passée comme une option de classe soit par `\KOMAOPTIONS` ou par `\KOMAOPTION` après le chargement de la classe, et que le paq de `typearea` est chargé et spécifié explicitement par `\usepackage` comme un argument optionnel lors de l'utilisation de `KOMA-Script`.

Les options `\KOMAOPTIONS` ou `\KOMAOPTION` changées après chargement du paq entraînent automatiquement le recalcul de la mise en page et des marges.

**DIV=*facteur***

L'option `\DIV=facteur` détermine la méthode de construction : la page est divisée horizontalement et verticalement en combien de bandes. La méthode exacte peut être trouvée [section 1.2](#). Il est important de savoir que n'importe quelle valeur entière peut être utilisée comme facteur à partir de 4, et que plus le facteur est grand, plus la zone de texte et les marges sont réduites. Il est bon de noter, cependant, que de très grandes valeurs peuvent mener à des conditions limites et nuire à une bonne méthode de construction, en fonction des autres options que vous aurez choisies. Ainsi, l'entête peut être, dans certains cas extrêmes, en dehors de la page. Alors, utiliser l'option `DIV=facteur` avec prudence, en conformité avec les principes de la ligne, dite elle-même, « typographique ».

Dans la [table 1.2](#) page 15, la taille de la page au format A4 sans correction de certains facteurs de `DIV` détermine la taille des mises en pages. La suite dépend des contraintes de taille de police.

Exemple : pour écrire un procès-verbal, vous choisissez d'utiliser la classe `protokol`, éditée en `recto-verso`, avec la police `Bookman` en 12 pt. Cette police est l'une des informations standard `PostScript` activée dans `LATEX`

avec la commande `\usepackage{bookman}`. La police `Bookman` s'étale, c'est-à-dire que ses caractères individuels sont très larges par rapport à sa hauteur. C'est pourquoi la valeur par défaut de `DIV` dans `typearea` est trop faible. Avec une valeur de 12 vous êtes au taquet. Cette étude, avec les sections suivantes comprises, est basée sur `DIV=15`. Le protocole n'est pas lié et une correction de reliure n'est pas nécessaire. Ainsi, vous écrivez :

```
\documentclass[A4paper,twoside]{protokol}
\usepackage{bookman}
\usepackage[DIV=15]{typearea}
```

Lorsque vous avez terminé, vous obtenez les journaux récents qui seront, à la fin du trimestre, tous liés ensemble comme un livre, aussi simplement qu'un collage. Pour cette liaison, y compris la reliure, un pli de 12 mm sera nécessaire, obligeant le changement de l'option `typearea` en conséquence et l'utilisation de la classe pour les protocoles selon la norme ISO 9000 :

```
\documentclass[A4paper,twoside]{iso9000p}
\usepackage{bookman}
\usepackage[DIV=15,BCOR=12mm]{typearea}
```

Vous pouvez aussi réutiliser une classe `KOMA-Script` :

```
\documentclass[twoside,DIV=15,BCOR=12mm]{scrartcl}
\usepackage{bookman}
```

L'option par défaut `A4paper` pourrait donc être supprimée.

Si vous utilisez `KOMA-Script` avec options de classe `\KOMAOPTIONS` ou `\KOMAOPTION` assurez-vous que `DIV` soit activé après le chargement de la classe et que `typearea` soit spécifié comme argument optionnel chargé par `\usepackage` pour que les recalculs soient automatiques.

<code>DIV=calc</code>
<code>DIV=classic</code>

Comme mentionné section 1.2, le format de papier A4 est un pré-réglage de valeur fixe pour `DIV`, présenté dans la table 1.2 page ci-contre. qui a l'inconvénient de ne pas prendre en compte le suivi de la police utilisée. Cela peut se produire si la police A4 est plutôt étroite et rendre très rapidement désagréable la lecture d'un nombre trop élevé de lettres par ligne. Si un format de papier différent est sélectionné, le calcul `typearea`

TABLE 1.2 – Valeurs de mise en page en fonction de DIV pour A4 excluant `topskip`

DIV	Typearea		Marges	
	largeur	hauteur	haut	intérieur
6	105,00	148,50	49,50	35,00
7	120,00	169,71	42,43	30,00
8	131,25	185,63	37,13	26,25
9	140,00	198,00	33,00	23,33
10	147,00	207,90	29,70	21,00
11	152,73	216,00	27,00	19,09
12	157,50	222,75	24,75	17,50
13	161,54	228,46	22,85	16,15
14	165,00	233,36	21,21	15,00
15	168,00	237,60	19,80	14,00

(toutes les longueurs sont en mm)

TABLE 1.3 – Paramètres prédéfinis de DIV pour A4

Taille de la police de base	10pt	11pt	12pt
DIV	8	10	12

offrira une bonne valeur de DIV. Vous pouvez choisir ce calcul pour A4 et utiliser `\DIV=calc` lieu de `\DIV=facteur`. Vous pouvez également spécifier explicitement cette option pour tous les autres formats de papier. Agissez de même si vous souhaitez utiliser le calcul automatique car il est possible de remplacer les valeurs par défaut dans un fichier de configuration (voir la [section 19.2](#)).

La méthode de construction classique mentionnée dans la [section 1.3](#) comme page d'un « livre canonique médiéval » est sélectionnable. Dans ce cas, au lieu de `DIV=facteur` ou simplement `DIV=calc`, utiliser `DIV=classic`. Il s'agit d'une valeur de DIV déterminée avec une bonne approximation possible de la page.

Exemple : les options répertoriées `DIV=facteur` posent exactement le problème d'une meilleure adéquation de la police, avec la fonte Bookman, en modifiant la valeur de `DIV`. Ainsi, dans cet exemple, il convient de déterminer cette valeur de `typearea` :

```
\documentclass[A4paper,twoside]{protokol}
\usepackage{bookman}
\usepackage[DIV=calc]{typearea}
```

Assurez-vous d'utiliser une des options de KOMA-Script ou paramétrer `\KOMAOPTIONS` après le chargement de la classe. Si l'option est changée via `\KOMAOPTIONS` ou `\KOMAOPTION` après le chargement du paq, la mise en page et les bordures sont automatiquement recalculées avec l'option `\DIV=classic`.

<code>DIV=current</code>
<code>DIV=last</code>

Si vous avez suivi les exemples, vous savez comment obtenir le calcul de `DIV` en fonction de la police sélectionnée. Le problème pourrait venir de la classe KOMA-Script qui charge par défaut le paq `typearea` rendant ainsi impossible la saisie des options comme arguments de `\usepackage`. Il n'est pas utile de préciser `DIV=calc` comme argument de `\documentclass` qui serait directement activée lors du chargement du paq `typearea`. La mise en page et les marges seraient calculées pour la police L<sup>A</sup>T<sub>E</sub>X en cours et ignorées pour le chargement de polices ultérieures. Calculer avec `\KOMAOPTIONS{DIV=calc}` la mise en page et les bordures d'un nouveau type de paq de police est possible après chargement, de même qu'avec `\KOMAOPTION{DIV}{calc}` dans lequel on précisera ensuite une valeur de `DIV` et de `calc`. Cependant, comme il est souvent plus pratique de mettre l'option `DIV` en place avant le chargement de police, dans un endroit bien en vue, par exemple en chargeant une classe, `typearea` propose deux autres valeurs symboliques pour cette option :

- avec `DIV=current`, l'actualisation simultanée de la mise en page et des bordures est lancée en utilisant exactement la valeur de `DIV` actuellement fixée, ce qui présente peu d'intérêt pour le recalcul de la mise en page si l'on choisit une police de base différente, mais qui, en revanche, est utile si vous voulez vous assurer de l'état limite du main-

- tien de la DIV par le changement de l'excédent `\text-height` moins `\topskip` qui doit représenter un multiple de `\baselineskip`,
- avec `DIV=last`, le recalcul de la mise en page et des bordures est lancé avec les paramètres utilisés dans le calcul précédent.

Exemple : pour une mise en page avec la police Bookman en utilisant, en même temps, une classe KOMA-Script, une bonne longueur de ligne doit être calculée. Rien de plus simple, on charge la valeur symbolique avec la commande `\KOMAoptions` :

```
\documentclass[BCOR=12mm{DIV=calc,twoside}{scr-artcl}
\usepackage{bookman}
\KOMAoptions{\cott{DIV=last}}
```

S'il est décidé d'utiliser plus tard une autre valeur de DIV, seul l'argument optionnel de `\documentclass` devra être changé. Pour un résumé de toutes les valeurs symboliques possibles pour l'option DIV, voir la table 1.4 page suivante. Il convient de noter que l'usage du paq `fontenc` peut amener L<sup>A</sup>T<sub>E</sub>X à utiliser une police différente.

Le recalcul est souvent nécessaire pour tenir compte de la variation de l'espacement des lignes étant donné que la mise en page doit être calculée strictement de telle sorte qu'un nombre entier de lignes entre dans la zone de texte qui devra elle-même être recalculée, ainsi que la zone de saisie normale pour l'interligne.

Supposons que, pour une thèse, la taille de police est de 10 pt avec une phrase demi-ligne requise. L<sup>A</sup>T<sub>E</sub>X, normalement fixé à 10 pt, ajoute 2 pt, à savoir 1,2 ligne. Il doit donc utiliser un facteur supplémentaire d'étirement de 1,25. Supposons qu'une correction de reliure de 12 mm soit également nécessaire. La solution à ce problème pourrait être la suivante :

```
\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]{scrreprt}
\linespread{1.25}
\KOMAoptions{\DIV=last}
```

`typearea` exécute toujours la commande `\normalsize` pour calculer une nouvelle mise en page.

Le même exemple à l'aide du paq de `setspace` (voir [FT11]) comme suit :

```
\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]{scrreprt}
\usepackage[onehalfspacing]{setspace}
```

TABLE 1.4 – Valeurs symboliques possibles pour DIV option de déclaration ou DIV argument

<code>areaset</code>	Réorganise la mise en page.
<code>calc</code>	Recalcule la mise en page et déterminer la valeur de DIV pour effectuer à nouveau.
<code>classic</code>	Recalcule pour effectuer encore après la page du livre canonique médiéval (calcul du cercle circonscrit).
<code>current</code>	Recalcule avec la valeur en cours de validité pour DIV.
<code>default</code>	Recalcule pour effectuer le passage à la valeur par défaut de la taille de la page en cours et de la police courante. Si aucune valeur par défaut n'existe, appliquer calc.
<code>last</code>	Recalcule avec le même DIV de l'argument donné à la dernière communication.

`\KOMAoptions{\DIV=last}`

Comme vous pouvez le voir dans l'exemple, le paq `setspace` ne s'applique qu'aux « pt » pour une police par défaut de tailles 10, 11 et 12 pt. Pour toutes les autres tailles de police, le paq utilise une approximation. À ce stade, il semble opportun de souligner que l'espacement des lignes de la page de titre doit être remis à la normale et les répertoires réglés avec l'interligne normale.

Un exemple complet serait :

```
\documentclass[10pt, twoside, BCOR=12mm, DIV=calc]{scrreprt}
\usepackage[onehalfspacing]{setspace}
\AfterTOCHead{\singlespacing}
\KOMAoptions{\DIV=last}
\begin{document}
\title{Titre}
\author{Papi Ray}
\begin{spacing}{1}
\maketitle
\end{spacing}
\tableofcontents
```



```
\chapter{Ok}
{document}
```

Voir également les notes de la section 2.8. La commande `\AfterTOCHead` est présentée dans la partie II, chapitre 14.

TABLE 1.5 – Argument BCOR symbolique possible

<code>current</code>	Recalcule à nouveau avec la valeur BCOR en cours de validité
----------------------	--

Rappelez-vous que ces options sont disponibles pour une utilisation avec `\KOMAOPTIONS` ou `\KOMAOPTION` après le chargement du paq et qu’elles déclenchent un recalcul automatique de la mise en page du texte et des marges.

<pre>\typearea[BCOR]{DIV} \recalctypearea</pre>
---

Si les options `DIV` ou `BCOR` accompagnent `\typearea`, la commande `\typearea` est appelée en interne. La valeur utilisée lors de l’activation de l’option `DIV` pour `BCOR` peut être trouvée dans la table 1.2 page 15 et si vous voulez recalculer la mise en page et les bordures de type pour `DIV` avec la valeur symbolique courante, vous pouvez utiliser directement `\typearea[current]{current}`.

Si les valeurs `BCOR` et `DIV` sont modifiées, l’utilisation de `\typearea` est recommandée, car dans ce cas les bordures et les niveaux de réglage sont recalculés en une seule fois.

Enfin, `\KOMAOPTIONS{DIV=DIV, BCOR=BCOR}` calcule les marges et établit de nouveaux niveaux pour `DIV` puis pour `BCOR`.

La commande `\typearea` est définie de sorte qu’il est possible de changer la mise en page d’un document en cours de création. Dans ce cas, les calculs sont faits sur une structure de base  $\LaTeX$  des valeurs internes et seuls les changements nécessaires sont réalisés pour éliminer les risques d’erreurs.

Il est probable que ces fonctions figurent dans les futures versions de  $\LaTeX$ , mais sans aucune garantie. Leur utilisation nécessite toujours, dans un document, une pagination depuis `\KOMAOPTION{DIV}{last}` ou depuis `\KOMAOPTIONS{\cotti{DIV=last}}` ou encore `\typearea[current]{last}` qui sont fréquemment nécessaires pour le calcul de la mise en page et des

marges, ce qui peut être considéré comme un raccourci de la commande `\recalctypearea`.

Si vous écrivez `\KOMAoptions{DIV=last}` ou `\typearea[current]{last}`, le recalcul de la mise en page et des marges est trop lourd en raison des nombreux caractères spéciaux, utiliser simplement `\recalctypearea`.

`twoside=valeur marche/arrêt`  
`twoside=semi`

Comme expliqué dans la section 1.2, la mise en page nécessite de savoir si un document doit être placé sur un ou sur deux côtés.

Dans un ensemble unilatéral, les marges gauche et droite sont de même largeur. En **recto-verso**, le bord intérieur d'une page est moitié moins grand que son bord extérieur respectif. Cette distinction, `twoside`, est une option `typearea` qui indique que le document est situé sur les deux côtés de la feuille. Les valeurs marche-arrêt dans la table 1.6 peuvent être utilisées comme valeurs par défaut. Si une option est utilisée sans indication de valeur, c'est la valeur `true`, **recto-verso**, qui sera utilisée par défaut. Désactiver l'option et vous avez un document à une seule face.

TABLE 1.6 – Valeurs KOMA-Script standard pour de simples interrupteurs

valeur	action
<code>true</code>	active l'option
<code>on</code>	active l'option
<code>yes</code>	active l'option
<code>false</code>	désactive l'option
<code>off</code>	désactive l'option
<code>no</code>	désactive l'option

Outre ces valeurs, la valeur `semi` peut être donnée. Elle entraîne un double ensemble de bords unilatéraux, avec des marges non alternatives simple face, ou une composition double face avec des marges à une face, et des notes possibles en marge. Une correction de liaison est possible depuis KOMA-Script 3.12 (voir options BCOR) qui ajoute, sur le bord gauche des pages impaires et sur le bord droit des pages paires, une mise en page **recto-verso**. En utilisant la compatibilité avec les versions antérieures

de KOMA-Script (voir section 1.6), le lien sera positionné dans la marge de gauche sur les deux pages avec l'usage de `twoside=semi`.

L'option peut être utilisée comme option de classe dans `\documentclass`, comme option de paq avec `\usepackage`, ou même après le chargement de l'ensemble de `typearea` avec `\KOMAOPTIONS` ou `\KOMAOPTION`. Après le chargement des paqs de `typearea`, elle permet le recalcul de la zone de type en utilisant `\recalctypearea`.

La composition `recto-verso` active avant la modification permet de créer un saut de page qui précédera la prochaine page impaire.

`twocolumn=`*valeur marche/arrêt*

Pour le calcul des types de zones avec `\DIV=calc`, savoir si le document comporte une ou deux colonnes est impératif. Les considérations sur la longueur de ligne de la section 2.1 s'appliquent alors individuellement à chaque colonne, un texte mis en page sera deux fois plus large avec une seule colonne.

L'option `twocolumn` de `typearea` indique que la page comporte un texte aligné sur deux colonnes. Une valeur marche-arrêt par défaut peut être utilisée pour de simples interrupteurs dans la table 1.6 page ci-contre. Si l'option est utilisée sans valeur spécifiée, la valeur `true` est supposée, elle indique le double colonnage. L'option désactivée indiquera, par défaut, une seule colonne.

L'option peut être réglée comme une option de classe `\documentclass`, comme une option de paq dans `\usepackage` ou après chargement de type zone de texte `\KOMAOPTIONS` ou `\KOMAOPTION`. L'utilisation de cette option après le chargement `typearea` entraîne automatiquement le recalcul de la mise en page par `\recalctypearea`.

`headinclude=`*valeur marche/arrêt*  
`footinclude=`*valeur marche/arrêt*

Jusqu'à présent, nous avons expliqué la construction de la mise en page en détaillant le contenu, la proportion entre bordures et contenu, mais une question cruciale reste posée : où se situe exactement la frontière entre les différents composants et que savons nous réellement de KOMA-Script ?

À première vue, cette question parait triviale. Les frontières sont dans les pages, l'entête qui reste au-dessus, les marges gauche ou droite, le contenu,

le pied. Mais ce n'est qu'une partie de l'histoire. Le bord extérieur n'est pas toujours vide car dans certains cas, il pourra contenir des notes marginales (voir la commande `\marginpar`, dans [DGS+12] ou l'article 2.21). Pour le haut et le bas, l'important est de savoir de quelle façon sont traités l'entête et le pied de page. S'agit-il d'entités gérées d'une manière indépendante ou faut-il tenir compte du format et des bords respectifs ? Répondre à cette question n'est pas facile, mais il semble probable qu'un pied de page et un entête vides soient tangentiels et ne seront pas isolés lors de la composition de la page. L'entête ou le pied de page qui ne contient que la pagination marquera une limite, sans impact sur l'effet visuel, et sera identifié puis reconnu comme tel lors de la lecture. Il devrait apparaître optiquement plus comme une marge et être inclus dans le présent. Il n'est pas pertinent, pour ce même effet visuel, de voir si le pied peut être facilement reconnu, lors de la lecture ou du survol, comme un pied ou pas. Ce qui importe est la façon dont l'empage bien rempli s'affiche avec la mise au point, lorsqu'on le regarde, ou comment il peut entraîner une visualisation floue.

Une suggestion : demander à vos grands-parents presbytes (en volant leurs lunettes) de lire la page à environ un demi mètre de la pointe de leur nez, ou bien essayer de lire la page tenue à bout de bras avec vos propres yeux, mais en n'en gardant qu'un seul ouvert. Les binoclards ont ici un net avantage : un pied de page qui contient non seulement le numéro, mais aussi des détails circonstanciels tels qu'un avis de droit d'auteur, ressemble plus, à distance, à une partie de corps de texte. Cet effet doit être pris en compte dans le calcul de la zone de saisie.

Pour l'entête, c'est un peu plus compliqué. Il contient fréquemment le titre de la colonne. Dans le cas d'une création de rubriques avec des titres infinis de chapitres et de sections, les lignes d'entête seront très longues et sembleront faire partie du corps de texte. Cet effet devient encore plus significatif lorsque l'entête contient non seulement le chapitre ou le titre de **section**, mais aussi le numéro de page.

Avec du texte sur les côtés droit et gauche, l'entête n'apparaîtra plus comme une marge vide, ce qui est moins évident avec le pied. La longueur des titres peut varier de sorte que l'entête apparaisse dans la marge sur une page et le texte sur la page suivante. Pourtant, ces pages ne doivent en aucun cas être traitées différemment, ce qui conduirait l'entête à sauter verticalement. Dans ce cas, il est préférable de l'inclure dans le cadre du

texte. La décision est facile lorsque le texte, l'entête ou le pied de page sont séparés par une ligne. Cela donne un aspect fermé et l'entête ou le pied de page semblent faire partie du corps de texte. Rappelez-vous : il importe peu que la ligne améliore la séparation optique du texte, seule l'apparence de la mise au point lorsque l'on regarde est importante. Le paq `typearea` ne peut pas prendre la décision de calculer séparément et de savoir s'il faut, ou ne faut pas, inclure entête et pied de page dans le cadre du corps de texte ou de la marge. Elles comprennent, pour de simples interrupteurs, les valeurs standard indiquées dans la table 1.6 page 20.

Comme je l'ai écrit ci-dessus, améliorer grâce à une ligne de séparation qui isolerait l'entête ou le pied de page n'est pas pertinent ici. Le facteur décisif est l'observation floue par le `typearea` qui vous aide à prendre la décision de les intégrer au corps de texte, séparément, en tant qu'options par les commandes `headinclude` et `footinclude`. On peut les utiliser sans spécifier une valeur et dans ce cas, imputer la valeur `true` comme valeur réelle marche-arrêt pour l'interrupteur simple. Pour mémoire, l'entête ou le pied de page sont inclus dans le cadre de la version anglaise du présent ouvrage.

Si vous n'êtes pas sûr d'un réglage correct, rappelez-vous que le réglage par défaut est généralement `headinclude=false` et `footinclude=false`. Cependant, lors de l'utilisation d'autres paqs, cela peut changer avec les classes `KOMA-Script` (voir la section 2.1 et le chapitre 5). Vous devez vous rappeler que ces options doivent être précisées comme options de classe par `\KOMAoptions` ou `\KOMAoption` après le chargement de l'une des classes et qu'un changement après le chargement de `typearea` ne conduit pas à un recalcul mais prend effet lors de la nouvelle mise en page. Pour cette opération, voir l'option `DIV` avec les valeurs courantes ou la commande `\recalctypearea`.

Outre les pages dont l'entête et le pied appartiennent à la zone de texte qu'ils délimitent (empagement), on rencontre des documents qui affichent des notes marginales intégrées avec l'option `mpinclude`. Elles peuvent prendre une valeur standard comme il est indiqué dans la table 1.6. Vous pouvez également ne pas préciser l'option d'affectation et dans ce cas, `true` est utilisé comme valeur de marche-arrêt.

`mpincluder=`*valeur marche/arrêt*

L'effet de `mpincluder=true` est que la zone de texte est réduite d'une unité de largeur qui sera utilisée pour gérer l'espace réservé aux notes marginales. Avec `mpincluder=false`, valeur par défaut, une partie sera utilisée pour les notes de marge. Ce réglage, d'une unité en recto ou d'une moitié en recto-verso n'est pas recommandé et son usage reste réservé, en général, aux experts. Dans la plupart des cas, l'option `mpincluder` est utile pour gérer un plus grand nombre de notes de marge, lorsqu'une partie de celles-ci doit être attribuée à la zone de texte. C'est par exemple le cas lorsqu'une ligne est utilisée pour une citation.

D'une manière habituelle, ces citations comportent un positionnement du corps de texte déjà réglé, mais comme l'empreinte n'est pas fermée, les extrémités de flottement peuvent déborder partiellement dans la marge. On change la possibilité d'utiliser `mpincluder` en augmentant la longueur `\marginparwidth` après le calcul de la zone de type ou encore, pour élargir, en utilisant l'expression `\addtolength-valeur` nécessitant une certaine finesse à l'usage. Voilà pourquoi cette option est conseillée aux experts. Bien sûr, vous pouvez étendre la largeur de la marge, par exemple d'un tiers sur le bord, en procédant comme suit pour obtenir l'effet désiré :

```
\setlength{\marginparwidth}{1.5\marginparwidth}
```

Comme il n'existe actuellement aucune option pour fournir plus d'espace pour les notes de marge dans la zone de texte, il n'y a qu'une seule façon de le faire : adapter `\textwidth` et `\marginparwidth` après le calcul de la zone type. Voir aussi `\AfterCalculatingTypearea` à l'article 19.1.

Vérifier que l'option est passée en utilisant une des classes KOMA-Script comme une option de classe ou par `\KOMAOPTIONS` ou `\KOMAOPTION` après le chargement de la classe. Changer l'option après le chargement `typearea` ne conduit pas à un calcul automatique, il ne concerne que la prochaine mise en page. Pour un nouveau calcul, charger l'option DIV avec les valeurs courantes ou charger la commande `\recalctypearea`.

`headlines=`*nombre de lignes*  
`headheight=`*hauteur*

Nous avons vu comment configurer une mise en page en utilisant le paquet `typearea` et comment spécifier si entête et pied de page font partie du

texte ou des marges. Pourtant, il manque encore quelques précisions, par exemple l'indication du nombre de lignes et la hauteur de l'entête utilisée pour les titres. Ces résultats sont obtenus avec les options `headlines` et `headheight`.

L'option `headlines` indique le nombre de lignes de l'entête. En principe, `typearea` fonctionne avec la valeur 1,25 par défaut qui est un compromis assez grand pour fournir l'espace à un entête souligné (voir [section 1.1](#)) et assez petit pour ne pas grignoter la bordure lors de l'utilisation d'un simple travail avec un entête non souligné. Ainsi, la valeur de la manchette par défaut conviendra dans la plupart des mises en page mais vous devrez l'adapter dans certains cas particuliers.

Si l'entête avec un texte comporte deux lignes cela signifie que, sur chaque côté, un avertissement de trop plein `overfull\ vbox` sera délivré par  $\LaTeX$ . Pour l'éviter `typearea` calculera une mise en page appropriée :

```
\documentclass[A4paper]{article}
\usepackage[headlines=2.1]{typearea}
```

Il est possible et même recommandé lors de l'utilisation d'une classe `KOMA-Script` de passer cette option directement à la classe :

```
\documentclass[headlines=2.1]{scrartcl}
```

Les commandes définissant le contenu des deux lignes de l'entête peuvent être trouvées dans le chapitre 5.

Dans certains cas, il est meilleur de spécifier directement la longueur et la hauteur de l'entête en tant que valeurs, ce qui est possible à l'aide de l'option alternative `headheight` dont  $\LaTeX$  connaît, par défaut toutes les longueurs et les tailles possibles. Cependant, pour mieux maîtriser votre présentation, il est préférable de paramétrer une option telle que `\baselineskip` au moment du calcul de la mise en page et des bordures. Rappelez-vous que ces options doivent être données comme options de classe soit par `\KOMAOPTIONS` ou par `\KOMAOPTION` après le chargement de classe en utilisant l'une des classes `KOMA-Script`.

Un changement de ces options après le chargement `typearea` ne conduit pas à un calcul automatique mais affecte plutôt le changement jusqu'au nouveau calcul de la prochaine mise en page. Pour connaître les modalités

de recalcul de la mise en page, voir l'option `DIV` avec les valeurs courantes ou la commande `\recalctypearea`.

```
footlines=nombre de lignes
footheight=hauteur
\footheight
```

De même que pour l'entête, la hauteur du pied n'est jamais indiquée, il faut donc en préciser les options. Par conséquent `typearea` définit pour la mise en place une nouvelle hauteur et une nouvelle longueur du pied si elles n'existent pas. Elles seront ensuite utilisées, par exemple, dans les classes et les paqs pour la conception de l'entête et du pied en fonction des classes et des paqs utilisés. Le faisceau KOMA-Script `scrlayer-scrpage` prend en compte la hauteur du pied de page `\footheight`, contrairement à `scrpage2`, et travaille donc activement avec `typearea`. Néanmoins, les classes KOMA-Script ne reconnaissent pas `\footheight` qui, sans le paq assistance, ne fournit que des styles de page avec une seule ligne dans le pied de page.

Comparable à celle de `headlines`, `footlines` est le rapport de l'option hauteur du pied de page, et normalement `typearea` fonctionne avec 1,25. Cette valeur est un compromis assez grand pour fournir l'espace pour un pied surligné ou souligné (voir section 1.1), et d'autre part assez petit pour ne pas empiéter sur la bordure quand on opère avec un pied de page simple sans ligne de séparation. Ainsi, la valeur par défaut dans la plupart des cas types est une bonne valeur, mais il vous faudra parfois régler pour des besoins ponctuels, comme pour l'entête, la hauteur du pied avec plus de précision.

Exemple : dans le pied, une déclaration de droits d'auteur doit être réglée sur deux lignes. Bien que cette fonction soit présente par défaut dans L<sup>A</sup>T<sub>E</sub>X, ce dernier ne teste pas l'espace affecté pour les pieds, ce qui peut entraîner un déséquilibre dans la répartition de la mise en page et des bordures. L'ensemble `scrlayer-scrpage` qui gère le contenu pourrait être pré-réglé après examen minutieux de cette même mise en page, ce qui rendra nécessaire, lors de son calcul, de préciser une plus grande hauteur de pied :

```
\documentclass[A4paper]{article}
\usepackage[footlines=2.1]{typearea}
```



Il est également possible de passer cette option directement en utilisant une classe KOMA-Script, `\documentclass[footlines=2.1]{scrartcl}`. Les commandes définissant le contenu du pied de page de deux lignes peuvent être trouvées dans le chapitre 5. Dans certains cas, il est utile de spécifier directement en tant que valeur de la longueur, la hauteur de la base, ce qui est possible à l'aide de l'option `\footheight`.

Il convient cependant de noter que l'utilisation d'une longueur de L<sup>A</sup>T<sub>E</sub>X comme `\baselineskip` demande un réglage de l'option indispensable au moment du calcul de la mise en page et des bordures.

Comme d'habitude, rappelez-vous que ces options doivent être données comme options de classe par `\KOMAOPTIONS` ou `\KOMAOPTION` après avoir chargé une classe en utilisant KOMA-Script. Un changement de ces options après le chargement de `typearea` ne mène pas à un calcul automatique de la mise en page, le changement ne concerne que le calcul de la prochaine mise en page. Voir la commande `\recalctypearea` ou l'option DIV avec les valeurs courantes.

```
\areaset [BCOR] {largeur}{hauteur}
```

Jusqu'à présent, beaucoup de choses ont été écrites ici sur la façon de créer une bonne mise en page pour une application standard que `typearea` peut rendre plus facile à l'utilisateur, tout en influençant dans le même temps son utilisation. Il existe aussi des cas où l'empage sera exactement maintenu à une certaine taille sans s'occuper de la conception de la mise en page ni de contraintes supplémentaires, les bordures réparties autant que possible pour le mieux et le cas échéant avec la prise en compte d'une reliure de correction.

L'option `typearea` propose `\areaset` pour remplir cette fonction, avec les paramètres largeur et hauteur de l'empage (zone de texte) à côté de la correction de reliure. Les bordures sont calculées automatiquement en prenant en compte les paramètres entête et pied de page des options `\headinclude` et `\footinclude`, mais les options `headlines`, `headheight`, `footlines` et `footheight` sont ignorées dans ce cas. Voir les informations supplémentaires sur `\areaset` dans la section 19.1.

La valeur par défaut de BCOR est 0pt, mais dans mon option BCOR, vous devez utiliser la valeur courante symbolique comme argument optionnel pour régler la correction de liaison,

Exemple : un texte, de type article imprimé sur du papier A4, avec une largeur exacte de 60 caractères, utilise la police Typewriter et 30 lignes par page. Vous pouvez travailler avec le préambule suivant :

```
\documentclass[A4paper, 11pt]{article}
\usepackage{typearea}
\newlength{\CharsLX} \% Largeur de 60 caractères
\newlength{\LinesXXX} \% Hauteur de 30 lignes
\settowidth{\CharsLX}{\texttt{1234567890}}
\setlength{\CharsLX}{6\CharsLX}
\setlength{\LinesXXX}{\topskip}
\addtolength{\LinesXXX}{29\baselineskip}
\areaset{\CharsLX}{\LinesXXX}
```

Le facteur de 29 au lieu de 30 est justifié par le fait que la base de la première ligne est déjà sur le bord supérieur de la mise en page aussi longtemps que la hauteur de la rangée du haut est inférieure à `\topskip`. La ligne supérieure ne nécessite aucune hauteur, la ligne inférieure du projet marque la limite de la mise en page.

Si au contraire vous voulez un livre de poésie avec une zone carrée de texte de longueur de côté de 15 cm et une marge de reliure de 1 cm, ce qui suit est conseillé :

```
\documentclass{poesie}
\usepackage{typearea}
\areaset[1cm]{15cm}{15cm}
\areaset[actuel]{\textwidth}{\textheight}
```

#### DIVareaset

La mise en page actuellement sélectionnée peut être réalignée en utilisant l'option `DIV=areaset` de la déclaration `\KOMAOPTIONS{DIV=areaset}`

```
\areaset[current]{\textwidth}{\textheight}
```

Le même résultat est obtenu avec `DIV=last`. La mise en page est alors définie par le dernier `\areaset`.

**Note :** *Le paq `typearea` n'a pas été fait pour gérer des valeurs prédéfinies de marge. Si vous devez le faire, utiliser le paq `geometry`.*

À ce propos, voir [Ume00].

## 1.8 Choix et réglage de la taille de papier

Le format de papier est une caractéristique de base indispensable à la création d'un document, comme la notion d'ensemble de la construction et de sa mise en page (voir les sections 1.1 à 1.4) avec le choix de la taille de papier. Alors que les classes `LATEX standard` sont limitées à quelques formats anglo-saxons, `KOMA-Script` a été créé pour gérer, avec le paq `KOMA-Script typearea`, les formats européens et même des pages avec des tailles inhabituelles.

```
paper=format paper=orientation
```

L'option de format du papier est l'élément central de sélection dans `KOMA-Script`, inspiré initialement du format légal d'une lettre américaine. En outre, le format ISO des lignes A, B, C et D est possible, par exemple A4 (ou a4 en minuscule). Les formats sont pris en charge avec la valeur indiquant figure ou paysage en option, ces valeurs ne diffèrent que par les programmes `dvips` qui s'articulent autour de  $-90^\circ$  pour le paysage tout en tournant à  $90^\circ$  pour une figure. Le format figure est surtout utile quand un programme `post-script` affiche des pages à l'envers en mode paysage. Ainsi, la différence joue un rôle et même l'option `pagesize` est une possibilité qui doit être utilisée.

Le format peut également prendre la largeur ou la hauteur de la forme, la hauteur devient la largeur en fonction de l'orientation du papier. Dans paysage avec `paper=landscape` ou marine avec `paper=seascape`, la plus petite dimension est la hauteur. C'est l'inverse avec `paper=portrait`. La première valeur était toujours la hauteur jusqu'à la version 3.01a, mais à partir des versions 3.01b et 3.21a, la première valeur s'applique à la largeur et la seconde à la hauteur, détail à respecter lors de l'utilisation d'un paramètre de compatibilité approprié (voir la version de l'option `section 1.6`).

Si vous voulez une carte d'index au format ISO A8 imprimée paysage avec des bordures à la cote minimale, sans pied de page ni entête :

```
\documentclass{article}
\usepackage[headinclude=false,footinclude=false,%
paper=A8,paper=landscape]{typearea}
\areaset{7cm}{5cm}
```

```
\pagestyle{empty}
\begin{document}
\section{Définition de format du papier}
letter, legal, executive, a0, a1 \dots%
b0, b1 \dots \c0, c1 \dots\d0, d1 \dots
\end{document}
```

Il existe des cartes au format spécial (largeur : 5 cm hauteur : 3 cm) qui rendent possible l'encodage suivant :

```
\documentclass{article}
\usepackage[headinclude=false, footinclude=false,
paper=landscape,paper=5cm:3cm]{typearea}
\areaset{4cm}{2.4cm}
\pagestyle{empty}
\begin{document}
\section{Définition de format du papier}
\%letter, legal, executive, a0, a1 \dots%
b0, b1 \dots\c0,c1 \dots\d0, d1 \dots
\end{document}
```

Dans les classes `standard` de L<sup>A</sup>T<sub>E</sub>X, le format utilisé par défaut est le format lettre américain. KOMA-Script travaille avec du papier A4 orienté portrait.

Cette option doit être transmise en utilisant une classe de KOMA-Script comme option de classe avec `\KOMAOPTIONS` ou `\KOMAOPTION`. Même un changement de l'orientation ou de la taille du papier après le chargement `typearea` ne conduit pas à un calcul automatique de la zone de type, il ne concerne que le nouveau calcul de la prochaine zone de saisie. À ce propos, voir l'option `DIV` avec les valeurs courantes ou la commande `\recalctypearea`.

`pagesize=opérateur de sortie`

Les descriptions ci-dessus permettent de sélectionner le format du papier selon les normes adaptées à notre système européen. Elles ont seulement une influence sur la sortie, mais elles redéfinissent en les remplaçant les paramètres internes de L<sup>A</sup>T<sub>E</sub>X. La fonction `typearea` l'utilisera ensuite dans la distribution de la page, pour positionner les marges et la zone de texte.

La spécification format DVI qui permet une sortie directe au format DVI vers un langage d'imprimante de bas niveau comme PCL<sup>4</sup> ou ESC/P2<sup>5</sup> peut être faite à tout moment avant le formatage du papier. Ceci a peu d'importance, généralement, mais fixe le point 0 de référence DVI en haut à gauche. Traduit en langage PostScript ou PDF, ce point est ailleurs et la taille du papier doit être spécifiée dans le fichier de sortie, information manquante.

TABLE 1.7 – Pilotes de sortie pour l'option `pagesize=output`

<code>auto</code>	Si les registres pdfTeX spécifiques <code>\pdfpagewidth</code> et <code>\pdfpageheight</code> sont présents, le pilote de sortie activé est <code>pdftex</code> et le pilote utilisé est <code>dvips</code> .
<code>automedia</code>	Gestion des pilotes de la fonction de sortie automatique. Gère et fixe en plus les registres des médias spécifiques VTeX, <code>\mediawidth</code> et <code>\mediaheight</code> largeur et hauteur du support.
<code>false, no, off</code>	Le format de papier n'est pas signalé au pilote de sortie.
<code>dvipdfmx</code>	Le format de papier est décrit <code>\special{pageSize=<i>largeur, hauteur</i>}</code> dans le fichier DVI. Le nom du pilote de sortie provient du fait que le programme <code>dvipdfmx</code> inclut un commutateur de format de papier.
<code>dvips</code>	Lorsqu'il est utilisé dans le préambule du document, le format du papier est, par défaut, celui utilisé dans le document <code>\special{pageSize=<i>largeur, hauteur</i>}</code> . Depuis <code>dvips</code> , il n'y a plus de commutateur de taille ce qui nécessite un <code>hack</code> plutôt compliqué pour réaliser la modification, à utiliser avec précaution, à vos propres risques.
<code>pdftex</code>	Le format du papier est sur les registres spécifiques <code>pdftex</code> <code>\pdfpagewidth</code> et dans la série <code>\pdfpageheight</code> , ce qui est également possible à tout moment dans le document.

Pour résoudre ce problème, les pilotes respectifs utilisent un format de papier prédéfini que l'utilisateur peut changer soit au moyen d'une option

4. PCL est le langage utilisé par HP pour ses imprimantes à encre et laser.

5. ESC/P2, est le langage EPSON pour ses imprimantes à aiguilles, à encre et laser.

ou par spécification dans le fichier source  $\TeX$ . Lorsque vous utilisez le pilote DVI `dvips`, l'information ne peut être donnée que sous la forme d'une commande spéciale avec `pdf $\TeX$`  ou `V $\TeX$`  pour mettre en place deux dimensions.

L'option `pagesize` permet de régler le format du papier, pour lequel un pilote gère le document de sortie. Les pilotes de sortie pris en charge peuvent être trouvés dans la table 1.7 page précédente.

La valeur par défaut est `pagesize=false`. L'utilisation de l'option sous la forme `pagesize` sans valeur correspondant à la valeur `pagesize=auto`.

Exemple : vous voulez un document utilisé comme un fichier DVI, qui permette également de créer une version en ligne au format PDF. Le préambule pourrait par exemple commencer comme ceci :

```
\documentclass{article}
\usepackage[paper=A4, \texttt{pagesize}]{typearea}
```

`pdf $\TeX$`  est maintenant utilisé pour le traitement et la sortie PDF est activée. Les deux dimensions spéciales `\pdfpagewidth` et `\pdfpageheight` sont fixées en conséquence. Toutefois, un fichier DVI peut être généré si  $\LaTeX$  ou `pdf $\LaTeX$`  est lancé par une commande spéciale écrite au début de ce fichier. Il est recommandé de toujours spécifier l'option `pagesize`. En règle générale, la méthode sans pilote de sortie est idéale.

## 1.9 Conseils

Pour la présentation d'un travail écrit pendant ses études, rapport ou thèse, il existe de nombreuses dispositions régulièrement contournées telle une typographie établie contre toutes les règles et qui viole les principes les plus élémentaires.

La raison en est bien souvent l'incompétence typographique de ceux qui les publient. Elle peut se justifier à l'origine par l'utilisation d'une machine à écrire mécanique ou par l'usage d'un traitement de texte basique datant du début des années 1980, avec lesquels, l'un ou l'autre, il n'était pas possible de produire une sortie typographiquement correcte sans effort considérable, mais la typographie idéale existe-t-elle ?

Des règles ont été créées, réalisables sans peine et semblant permettre une correction facile. Elles comprennent des paramètres avancés utiles pour l'impression recto avec une machine à écrire, et dans le but d'éviter les interlignes courts composés par étourderie, les marges ont été maintenues étroites et l'espacement des lignes augmenté à 1,5 pour les corrections. Avant l'avènement des systèmes modernes de traitement de texte, la seule alternative à T<sub>E</sub>X aurait été l'interligne simple. Dans un tel document, même des signes de correction auraient été difficiles à ajouter. Lorsque l'ordinateur est devenu plus largement disponible pour traiter le texte, des étudiants ont utilisé de *belles polices*, pensant améliorer le visuel de leur travail. Il est facile d'oublier que ces polices sont pénibles à lire et donc impropres à ce nouvel usage. Deux types de corps de lettres devenus largement utilisés semblaient adaptés... time par exemple, police du début du 20<sup>e</sup> siècle relativement étroite, conçue pour les colonnes des journaux britanniques que l'on retrouve fréquemment prescrites dans les versions modernes, ne correspondant pas vraiment à une spécification de taille.

Le développement de L<sup>A</sup>T<sub>E</sub>X rend son usage de plus en plus fréquent, avec une possibilité accrue pour les corrections efficaces de lignes sensibles. Mais il est souvent difficile de mettre en œuvre des règles typographiques discutables ou délicates lorsque seul un nombre réduit en sera respecté lors de l'utilisation d'une police non-proportionnelle, dont il existe cependant quelques bons modèles. Un texte dans une écriture qui semble bonne peut parfois être amélioré, par exemple, avec une base en saillie, petit « i » ou « l » donnant une autre largeur de caractère, ou encore l'utilisation d'un empattement justifié typographiquement car compatible avec l'écriture habituelle de notre langue française. Certains paramètres peuvent être ignorés ou interprétés généreusement comme « utiliser 60 caractères par ligne » comme valeur moyenne ou maximale, non comme valeur fixe.

Les paramètres d'enregistrement sont généralement destinés à obtenir lors de leur exécution un résultat utile, même si l'auteur ne sait pas ce qui doit être observé. Utile signifie lisible et corrigé et à mon avis, un ensemble L<sup>A</sup>T<sub>E</sub>X avec l'option `typearea` pour la mise en page répond, dès le début, à ces exigences. Si vous êtes confronté à des règles sensiblement différentes, je vous suggère de tester la mise en forme de votre texte avant de fournir la version définitive de votre travail. En cas de besoin, il pourrait être ajusté

en modifiant la mise en page à l'aide de l'option `DIV\areaset`. L'auteur le déconseille et vous recommande de ne l'utiliser qu'avec précaution.

Dans le pire des cas, utiliser le paq `geometry` (voir [Ume00]), ne faisant pas partie de *KOMA-Script*, ou modifier le paramètre d'empage dans  $\LaTeX$  lui-même. Les valeurs définies par `typearea` et enregistrées dans le fichier `journal`<sup>6</sup> rendent des ajustements modérés possibles. Mais veiller à ce que les proportions de l'empage correspondent à peu près à celles de la page en cours de correction avec sa reliure. Si l'espacement du texte doit être, par exemple, redéfini d'une demi-ligne, n'utilisez pas `\baselinestretch`. Cette approche souvent recommandée est obsolète depuis l'introduction de  $\LaTeX$  en 1994. En désespoir de cause, utiliser l'instruction `\linespread`.

Je recommande néanmoins le paq `setspace` bien qu'il n'appartienne pas à *KOMA-Script* (voir [FT11]). Vous devrez recalculer la mise en page de `typearea` après la conversion de l'interligne et revenir à l'espacement de lignes normal pour le titre, pour le contenu des tables et listes diverses. Le paq `setspace` propose aussi ses propres instructions pour la gestion des environnements spécifiques bibliographie et index. Le paq `typearea` adapte, à l'aide de l'option `\DIV=calc`, une zone de texte généreuse et de nombreux typographes traditionalistes indiquent que la longueur de la ligne qui en résulte est encore trop grande. La valeur `DIV` pour chaque document figure dans le fichier `journal` et peut être modifiée sans difficulté après la première exécution de  $\LaTeX$ .

Il n'est pas rare que l'on me pose cette question :

« Pourquoi justifier une évaluation chapitre après chapitre ressemblant à une répétition rabâchée alors qu'il serait plus simple de consacrer un paragraphe à la manière d'ajuster les marges comme dans un traitement de texte? ».

J'ai souvent pensé que cette solution serait sans doute la meilleure mais tout le monde sait que, pour choisir la bonne marge, les bordures de *KOMA-Script* ne sont pas toujours la formule idéale.

Permettez-moi pourtant de conclure cette **section** par une citation de Hans Peter Friedrich WILLENBERG et de FORSSMANN, deux maîtres en typographie des plus distingués (voir [WF00]) :

---

6. Pour mémoire, il s'agit du fichier avec une extension `.log` dans votre dossier.



Faire soi-même est depuis longtemps l'habitude, une longue habitude, les résultats sont souvent discutables car les simples typographes ne voient pas ce qui est faux et ne peuvent pas savoir ce qui est important. Alors, vous vous habituez à une typographie douteuse, parfois mauvaise [...] Et pourtant, une génération pourrait venir qui reprendrait goût à la chose typographique, avec l'envie de l'appliquer à la décoration...



# Chapitre 2

## Les principales classes

Les principales classes KOMA-Script sont l'égal de leurs homologues des classes L<sup>A</sup>T<sub>E</sub>X standard. Cela signifie que KOMA-Script contient trois classes remplaçant les classes standard : `scrbook`, `scrreprt` et `scrartcl` (livre, rapport et article). La classe de document pour les lettres, tout à fait différente, est décrite à part. La façon la meilleure d'utiliser une classe KOMA-Script par défaut consiste à remplacer son nom dans la commande `\documentclass`.

Conformément à la table 2.1, remplacer `\documentclass{book}` par `\documentclass{scrbook}`, entraîne l'exécution automatique de L<sup>A</sup>T<sub>E</sub>X sans exigence ni disposition particulière.

Une grande partie des nombreuses possibilités et options proposées par les classes KOMA-Script est décrite dans les sections suivantes.

TABLE 2.1 – Correspondance entre classes standard et KOMA-Script

Classes Standard	Classes KOMA-Script
article	scrartcl
report	scrreprt
book	scrbook
letter	scrlttr2

Pour expliquer le principe des classes, souvent, vous hésitez au début d'un document, sur les options à choisir. Certains paramètres, tels que la sélection du format de papier, peuvent être fixés par défaut, mais la question de la mise en page peut être difficile à déterminer à l'avance. D'autre part, ces informations peuvent être tenues comme négligeables pour les principales activités de l'auteur : projet de plan, rédaction du texte, compilation d'images, de tables ou de répertoires. Pour l'auteur, il est préférable de se concentrer sur le contenu sans prêter attention aux subtilités de la mise en forme. Outre le choix des options, cela inclut aussi des éléments tels que la correction de la mise en page et les interventions possibles dans le positionnement, par exemple, des figures et tables.

### 2.1 Option de sélection précoce ou tardive

Appliquer *mutatis mutandis* ce qui a été écrit dans la **section 1.5**. Donc, si vous avez lu et compris cette partie, vous pouvez continuer **section 2.2**.

La présente décrit une caractéristique du programme qui, outre `scrbook`, `scrreprt` et `scrartcl`, affecte d'autres classes KOMA-Script et d'autres paqs. Pour permettre à l'utilisateur de découvrir toutes les informations sur un paq ou une classe, cette **section** se trouve presque à l'identique dans plusieurs chapitres. Pour qui veut obtenir une vue d'ensemble de KOMA-Script, elle peut être lue dans l'un des chapitres et ignorée dans les autres.

```
\documentclass[liste des options]{KOMA-Script-classe}  
\usepackage[liste des options]{liste des paqs}
```

Si L<sup>A</sup>T<sub>E</sub>X prévoit que les utilisateurs spécifient les options de classe sous la forme d'une simple liste de mots-clés séparés par des virgules comme argument optionnel de `\documentclass`, elles sont également transmises à tous les paqs qui comprennent ces options. De même, il est envisagé que les utilisateurs spécifient les options de paq sous la forme d'une simple liste des mots-clés comme argument optionnel de `\usepackage`. KOMA-Script étend le mécanisme des options pour les classes et quelques paqs à d'autres possibilités. Ainsi, la plupart ont une valeur supplémentaire, n'existent pas nécessairement que sous la forme d'option et peuvent aussi prendre

la forme `option=valeur`. Jusqu'à cette modification `\documentclass` et `\usepackage` travaillent avec KOMA-Script comme décrit dans [Tea05b] ou dans chaque introduction de L<sup>A</sup>T<sub>E</sub>X, par exemple [DGS12].

Si vous utilisez une classe KOMA-Script aucune option n'est spécifiée lors du chargement des paqs `typearea` ou `scrbase`. Les classes chargent ces paqs sans options mais L<sup>A</sup>T<sub>E</sub>X refuse de charger, à répétition, un paq qui contient différentes options. Avec `\documentclass` et `\usepackage` les paramètres des longueurs, compteurs, etc. peuvent facilement casser et doivent être attentivement surveillés. Ainsi, l'utilisation d'une option de longueur de L<sup>A</sup>T<sub>E</sub>X provoque dans ces instructions un message d'erreur avant que la valeur de KOMA-Script ne soit adoptée.

L'affectation de valeurs avec des longueurs ou des compteurs de L<sup>A</sup>T<sub>E</sub>X ne doit jamais précéder `\documentclass` ou `\usepackage`, mais toujours suivre les instructions documentées `\KOMAoptions` ou `\KOMAoption`.

```
KOMAoptions{liste des options}
KOMAoption{Option}{liste des valeurs}
```

KOMA-Script offre la possibilité de modifier la valeur des options dans la plupart des classes et des paqs.

Vous pouvez modifier cette valeur avec l'instruction `\KOMAoptions`, qui permet de la donner à chaque option de la liste.

Certaines options peuvent avoir une valeur par défaut (*DefaultValue* en anglais). Si la valeur d'option est omise, la valeur par défaut est appliquée automatiquement.

Certaines options peuvent contenir simultanément plusieurs valeurs. Il est alors possible de leur assigner, en option séquentielle, un ensemble de valeurs. Les valeurs individuelles sont, dans la liste de valeurs, séparées les unes des autres par des virgules. Si une valeur contient un signe égal ou une virgule, alors la valeur est mis entre accolades.

Pour la mise en application de cette commande, KOMA-Script utilise `\FamilyOptions` et `\FamilyOption` avec la famille « KOMA », voir dans la Partie II, la section 11.3.

## 2.2 Compatibilité avec les versions précédentes

Appliquer *mutatis mutandis*, ce qui a été écrit dans la section 1.5.

Archiver le code source de ses documents permet habituellement à L<sup>A</sup>T<sub>E</sub>X de fonctionner en obtenant toujours exactement le même résultat dans l'avenir. Mais dans certains cas, apporter améliorations et corrections au paq conduit à des changements de comportement, en particulier au cours d'une transition, ce qui est parfois indésirable.

<code>version=<i>valeur</i></code>
<code>version=<i>first</i></code>
<code>version=<i>last</i></code>

Depuis la version 2.96a le choix est de savoir si un fichier source doit, à l'avenir, conduire l'exécution de L<sup>A</sup>T<sub>E</sub>X exactement au même résultat, ou si vous souhaitez faire correspondre chaque ajustement à la dernière version de la classe en utilisant l'option compatibilité. Une prise en charge de la compatibilité avec une ancienne version de KOMA-Script peut tout d'abord se faire avec `version=first` ou `version=2.9` ou `version=2.9t`. Lorsque vous spécifiez une version inconnue en tant que valeur, un avertissement est émis et `version=first` adopté par défaut.

La `version=last` sélectionne la dernière version, la compatibilité future est omise. Si l'option est utilisée sans valeur, `last` est, et restera, la valeur par défaut, aussi longtemps qu'aucune option obsolète n'est utilisée.

Dans ce cas, KOMA-Script2 définit KOMA-Script3 automatiquement avec `version=first`. Un avertissement explique comment vous pouvez éviter cette commutation de compatibilité. Alternativement, vous pouvez choisir une version différente de l'option de réglage même après l'activation de l'option obsolète.

La question de la compatibilité concerne principalement la mise en page des questions en ligne. De nouvelles possibilités qui ne l'impactent pas sont disponibles même si une compatibilité avec une version plus ancienne est sélectionnée. L'option n'a pas d'effet sur les changements de transition qui résultent de l'élimination des erreurs manifestes lors de l'utilisation d'une version plus récente. En cas de besoin d'un texte compatible avec la mise en page de la version KOMA-Script, il est sage d'archiver le document utilisé par défaut.

Il est important de noter que la version de l'option ne peut pas être modifiée après le chargement de la classe, le réglage avec `\KOMAOPTIONS` ou `KOMAoption` n'est donc pas destiné à cette option,

## 2.3 Mode création

De nombreuses classes comptent un grand nombre de paqs en mode normal et en mode création. Les différences entre ces deux modes sont aussi variées que les classes et les paqs qui offrent cette distinction.

<code>draft=valeurs de marche-arrêt(on-off)</code>
--

Avec cette option, une distinction est faite entre les documents finaux et de projet. Une valeur de `marche-arrêt` peut être l'une des valeurs standard utilisée pour de simples interrupteurs de la table 1.6 page 20. Activée, elle permet à l'œil non averti de trouver un paragraphe à identifier et à traiter manuellement. En revanche le paramètre `draft=false` dans le projet n'affiche aucune boîte par défaut. Ces lignes disparaissent souvent en utilisant le paq `microtype` [Sch13].

## 2.4 Mise en page

Une page de document est composée de différentes parties telles que les marges, l'entête, le pied, l'empage (la zone de texte) et les distances entre ces éléments. `KOMA-Script` distingue également la page totale, le format du papier et le côté visible. La division et la reconnaissance des espaces appartiennent aux différentes compétences de base d'une classe. Ce travail est réalisé par le paq `typearea`, chargé automatiquement. Il n'est ni nécessaire, ni utile de l'appeler en utilisant une classe `KOMA-Script` par `\usepackage`. Il peut également être utilisé avec d'autres classes et nécessite alors d'être chargé. Certains paramètres des classes `KOMA-Script` ont un impact sur la mise en page. Ces effets sont documentés dans les paramètres appropriés en particulier pour les documents recto-verso. Pour plus d'explications sur le choix de la taille du papier, la division de la page en marges, la mise en page et le choix de la composition à une ou deux colonnes, consulter la documentation du paq `typearea`, chapitre 1.

<code>\flushbottom</code>
<code>\raggedbottom</code>

Si une page ne contient que du texte sans paragraphe ni rubrique, les règles de base sont automatiquement respectées à l’empage. Cependant, quand un paragraphe est décalé d’une demi-ligne de base, il suffit que le nombre de points sur un côté diffère du nombre de l’autre côté pour que l’objectif ne soit pas atteint. Comprimer ou étendre une partie de l’espace vertical devient nécessaire pour atteindre la destination.  $\text{T}_{\text{E}}\text{X}$  peut gérer, à cet effet, les distances extensibles et compressibles alors que  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  effectue automatiquement cet équilibre vertical. Si l’option `twoside` recto-verso est demandée, l’équilibre vertical est également activé (voir la [section 1.5](#)).

Vous pouvez utiliser `\flushbottom` à tout moment en l’appelant dans la page courante. L’inverse est possible avec `\raggedbottom` qui est la valeur pour l’enregistrement recto par défaut. Par ailleurs, `KOMA-Script` utilise un type légèrement modifié de l’abandon de réglage vertical. Cela a été fait pour déplacer les notes au bas de la zone de texte au lieu de les avoir près de la dernière ligne de texte utilisée.

## 2.5 Choix de la taille de police du document

La police de base et sa taille sont des éléments clés dans la conception d’un document, la division entre le type et la zone des bords en dépend essentiellement. La police de base est la fonte qui est utilisée pour la majeure partie du texte d’un document. Tous les paramètres déviants, que ce soit dans la forme, l’épaisseur, l’angle et la taille lui sont liés.

<code>fontsize=taille</code>
------------------------------

Alors qu’un nombre très limité de tailles de police est pris en charge par les classes `standard` et la plupart des autres classes, `KOMA-Script` offre la possibilité de spécifier une taille pour la police de base. Toute unité de  $\text{T}_{\text{E}}\text{X}$  connue peut être utilisée comme unité de taille. Si le format est défini sans unité, alors le `pt` est supposé être l’unité.

Si l’option `point` est définie dans le document, la taille de base de la police change pour toutes les variables dépendantes. Cela peut être utile



lorsqu'un réglage demande une taille de police plus petite. Il convient de noter qu'une modification après le chargement de la classe ne redémarre pas automatiquement la répartition calculée entre les niveaux de taux et les bords (voir `\recalctypearea`, section 1.2). Le nouveau calcul est cependant basé sur la taille de la police. Les effets du changement de la taille de la police de base peut donc entraîner des erreurs sur les paqs chargés et ne peut pas être considéré comme une erreur de KOMA-Script.

Cette option ne devrait jamais être interprétée comme un substitut pour `\fontsize` (voir [Tea05a]) et ne devrait donc jamais être utilisée à la place des instructions de la taille de police de base, de `\tiny` (minuscule) à `\Huge` (énorme).

La police est par défaut pour `scrbook`, `scrcrpt` et `scrartcl` `size=11pt`. Par contre, la police est prédéfinie à 10pt dans les classes `standard`. Cela est à considérer, si nécessaire lors de la modification des classes `standard` pour les classes KOMA-Script.

## 2.6 Description de textes

L<sup>A</sup>T<sub>E</sub>X comporte un certain nombre d'instructions de mise en page en plus du choix de la police, ce qui inclut également des commandes pour sélectionner la taille ou l'alignement du texte. Pour plus de détails sur les possibilités normalement définies, consulter et se référer à [DGS+12], [Tea05a] et [Tea05b].

<code>\textsuperscript{texte}</code> <code>\textsubscript{texte}</code>
--

Dans le noyau L<sup>A</sup>T<sub>E</sub>X se trouve prédéfini `\textsuperscript` qui peut être employé pour manipuler du texte. Malheureusement, L<sup>A</sup>T<sub>E</sub>X ne prévoit pas d'instruction appropriée pour placer un texte en indice et pour ce faire, KOMA-Script utilise donc `\textsubscript`.

Si vous rédigez un texte sur le métabolisme humain qui nécessite de temps en temps des formules chimiques avec de simples chiffres, en termes de balisage logique, vous définissez d'abord, dans le préambule de votre document ou dans un paq séparé, une nouvelle commande :

```
\newcommand*{\Molek}[2]{#1\textsubscript{#2}}
```

Puis vous écrivez :

La cellule produit son énergie de la réaction de \Molek C6  
\Molek H{12}\Molek O6 et \Molek O2 en \Molek H2 \Molek O{  
et \Molek C{ }\Molek O2. L'arsenic \Molek(As){} a un effet  
défavorable sur le métabolisme.

Le résultat se présente comme suit :

La cellule produit son énergie de la réaction de C<sub>6</sub>H<sub>12</sub>O<sub>6</sub> et  
O<sub>2</sub> en H<sub>2</sub>O et CO<sub>2</sub>. L'arsenic (As) a un effet défavorable  
sur le métabolisme.

Un peu plus tard vous décidez que les formules de somme devraient être rédigées **sans empatement**. Pour rendre la décision conforme au balisage logique, vous avez seulement besoin de redéfinir la déclaration \Molek :

```
\newcommand*{\Molek}[2]{%  
\textsf{#1\textsubscript{#2}}%  
}
```

qui produit les variations du texte des formules dans tout le document.

La cellule produit son énergie de la réaction de **C<sub>6</sub>H<sub>12</sub>O<sub>6</sub>** et  
**O<sub>2</sub>** en **H<sub>2</sub>O** et **CO<sub>2</sub>**. L'arsenic (**A<sub>s</sub>**) a un effet dévaforable sur  
le métabolisme.

En utilisant les commandes \setkomafont et \addtokomafont on peut modifier les caractéristiques d'un élément particulier. Théoriquement, les commandes peuvent être utilisées, y compris la sortie de texte, avec toutes les instructions possibles. Cependant, vous devez nécessairement limiter ces déclarations à un seul attribut de police.

```
\setkomafont{élément}{commande}  
\addtokomafont{élément}{commande}  
\usekomafont{élément}
```

En général, les commandes comme \normalfont, \rmfamily, \sffamily, \ttfamily, \mdseries, \bfseries, \upshape, \itshape, \slshape, avec \scshape ou l'une des commandes \Huge, \huge, \LARGE, \Large, \large, \normalsize, \small, \footnotesize, \tiny et \scriptsize concernent une police normale.

Pour le détail de ces commandes, reportez-vous à [DGS+12], [Tea05a] ou [Tea05b].

Les commandes de commutation de la couleur telles que `\normalcolor` sont possibles (voir [Car05] et [Ker07]). L'utilisation d'autres commandes, celles qui mènent par exemple à une redéfinition ou à une édition, n'est pas prévue, mais elle peut provoquer un étrange comportement qui ne doit pas être forcément considéré comme un défaut.

La substitution de police se fait par `\setkomafont` avec une redéfinition. En revanche, la définition existante est étendue avec `\addtokomafont`. Il est donc conseillé d'utiliser les deux états dans le préambule ainsi que de se référer aux exemples d'utilisation des différents éléments dont les noms et les significations sont énumérées dans la table 2.2 et les valeurs par défaut indiquées dans les paragraphes correspondants.

L'option `\usekomafont` permet la modification de la police actuelle et de redéfinir l'élément spécifié. Pour l'élément `captionlabel`, on utilise la même police que pour `descriptionlabel`, ce peut être précisé avec :

```
\setkomafont{captionlabel}{%
\usekomafont{descriptionlabel}%
}
```

D'autres exemples figurent dans les sections sur les différents éléments.

TABLE 2.2 – Éléments dont le type peut être modifié dans `scrbook`, `scrreprt` ou `scrartcl` avec `setkomafont` et `addtokomafont`

Éléments	Description
<code>author</code>	Auteur dans le titre principal en utilisant <code>\maketitle</code> et l'argument <code>\author</code>
<code>caption</code>	Texte d'une figure, d'une table ou d'un titre
<code>captionlabel</code>	Étiquette du sous-titre d'une figure, d'une table ou d'un titre avec application selon la légende de l'élément
<code>chapter</code>	Titre de la section unité <code>\chapter</code>
<code>chapterentry</code>	Niveau matières d'entrée <code>\chapter</code>

<code>chapterentrydots</code>	Points de connexion d'entrée dans la table des matières en option au niveau <code>\chapter</code> contrairement à <code>chapterentrypagenumber</code> (voir sect. 1.9)
<code>chapterentrypagenumber</code>	Numéro de page de la table de niveau d'entrée de contenu <code>\chapter</code> différent du chapitre élément entrée
<code>chapterprefix</code>	Ligne du numéro de chapitre intégrant les arguments <code>chapterprefix=true</code> ou <code>appendixprefix=true</code>
<code>date</code>	Date dans le titre principal du document avec <code>\maketitle</code> , si l'on paramètre l'argument <code>\date</code>
<code>dedication</code>	Dédicace après le titre principal du document avec <code>\maketitle</code> , si l'argument <code>\dedication</code> est actif
<code>descriptionlabel</code>	Étiquette d'argument optionnel de déclaration <code>\item</code> , dans un environnement de détail (voir section 3.18)
<code>dictum</code>	Avec <code>\dictum</code> la maxime sort plus finement
<code>dictumauthor</code>	Auteur d'une énonciation intelligente; application selon l'élément maxime
<code>dictumtext</code>	Autre nom pour maxime
<code>disposition</code>	Tous les titres de section, les arguments de <code>\part</code> jusque <code>\subparagraph</code> et <code>\minisec</code> , y compris le résumé utilisé avant l'élément de niveau correspondant
<code>footnote</code>	Marque et texte d'une note de pied de page
<code>footnotelabel</code>	Marque d'une note de pied de page selon l'élément <code>note</code>
<code>footnotereference</code>	Référence à la note de pied de page dans le texte
<code>footnoterule</code>	Ligne au-dessus des notes
<code>labelinglabel</code>	Étiquette, argument optionnel de la commande <code>\item</code> , et séparateur, c'est-à-dire argument optionnel de l'environnement d'étiquetage

<code>labelingseparator</code>	Séparateur, argument optionnel d'un environnement en matière d'étiquetage et dans cet environnement, application selon l'élément <code>labelinglabel</code>
<code>minisec</code>	Rubrique avec l'ensemble des titres <code>\minisec</code>
<code>pagefoot</code>	N'est utilisé que si les paqs <code>scrpage2</code> ou <code>scrpage2</code> sont chargés
<code>pagehead</code>	Autre nom pour le pied de page ( <code>pageheadfoot</code> )
<code>pageheadfoot</code>	Entête et pied de page dans tous les styles définis de page <code>KOMA-Script</code>
<code>pagenumber</code>	Numéro de page dans l'entête ou le pied de page
<code>pagination</code>	Autre nom pour le numéro de page
<code>paragraph</code>	Titre du niveau <code>\paragraph</code>
<code>part</code>	Titre du niveau <code>\part</code> , mais sans numéro de ligne
<code>partentry</code>	Niveau d'entrée de la table des matières <code>\part</code>
<code>partentrypagenumber</code>	Numéro de page d'entrée de la table des matières <code>\part</code> différente de l'entrée des éléments
<code>partnumber</code>	Numéro de la partie dans le répertoire <code>\part</code>
<code>publishers</code>	Informations éditeur dans le titre principal du document avec <code>\maketitle</code> , si l'argument <code>\publishers</code> est utilisé
<code>section</code>	Titre du niveau <code>\section</code>
<code>sectionentry</code>	Niveau d'entrée matières <code>\section</code> (disponible seulement pour <code>scrartcl</code> )
<code>sectionentrydots</code>	Points de connexion optionnel du contenu <code>\section</code> de la table des matières accessible depuis <code>sectionentrypagenumber</code> (exclusivement avec <code>scrartcl</code> , voir <code>section2.9</code> )
<code>sectionentrypagenumber</code>	Nombre de pages de contenu entrée <code>section</code> en dépit de l'article entrée de l'élément répertoire (uniquement disponible pour <code>scrartcl</code> )

<code>sectioning</code>	Autre nom pour la disposition
<code>subject</code>	Sujet du document dont l'argument <code>\subject</code> est en relation avec la page de titre complète avec <code>\maketitle</code>
<code>subparagraph</code>	Titre du niveau <code>\subparagraph</code>
<code>subsection</code>	Titre du niveau <code>\subsection</code>
<code>subsubsection</code>	Titre du niveau <code>\subsubsection</code>
<code>subtitle</code>	Sous-titre du document, incluant l'argument <code>\subtitle</code> de la page de titre principal <code>\maketitle</code>
<code>title</code>	Titre principal du document, avec l'argument <code>\title</code> lors de l'utilisation de <code>\maketitle</code> (par rapport à la taille du titre principal, voir, dans le texte de l'article 2.7) les remarques supplémentaires
<code>titlehead</code>	Entête au-dessus du titre principal du document, avec l'argument <code>\titlehead</code> lors de l'utilisation de <code>\maketitle</code>

<pre> \usefontofkomafont{élément} \useencodinguikomafont{élément} \usesizeofkomafont{élément} \usefamilyofkomafont{élément} \useseriesofkomafont{élément} \useshapeofkomafont{élément} </pre>
---

Bien que vivement déconseillé car ils n'ont rien à voir avec l'écriture, il arrive que certains paramétrages soient appliqués pour l'utilisation d'une police. Pour modifier le réglage d'une police, utiliser uniquement les paqs `\usekomafont`, `\usefontofkomafont` ou `\usesizeofkomafont` pour à la fois régler la taille de la police et la distance de base de référence. Cette instruction suppose que la taille de la police et de la distance de base, la famille, la largeur de ligne et de série, sa forme ou son orientation et l'encodage peuvent être acceptés avec le reste des instructions et des attributs individuels.

## 2.7 Titre du document

Permet de distinguer entre deux types de titres pour un ensemble de documents. Il y a d'une part, la ou les pages de titre : c'est le titre du document ainsi que des informations supplémentaires, tel que l'auteur. En plus du titre, la page principale peut contenir un `demi-titre`, les informations de l'éditeur ou des remerciements. Ci-dessous le titre en ligne d'un document qui continue, par exemple, avec le résumé, une préface ou la table des matières.

```
titlepage=valeur de marche/arrêt
titlepage=première de couverture
```

Cette option est sélectionnée si `\maketitle` est utilisé, ou défini comme option de titre. Voir table 1.6 page 20 comment une valeur de marche-arrêt peut être utilisée comme valeur standard pour de simples interrupteurs. La forme de page de titre est sélectionnée par `titlepage` ou `titlepage=true`. La commande `\maketitle` utilise l'environnement `titlepage` pour définir un site qui ne reçoit ni entête ni pied de page. Comparée avec les classes `standard`, `KOMA-Script` propose une offre de fonctions élargie. En revanche, `titlepage=false` n'assure qu'une rubrique : le titre est réglé en anglais. Autrement dit, le préreglage soulignera le titre mais pourra mettre en évidence un résumé ou un article définis. Avec la troisième option, `titlepage=firstiscover`, ce ne sont pas que les pages de couverture qui sont activées. On s'assure également que `\maketitle` permet de sortir le `demi-titre` ou le titre principal comme une page de couverture. Ce réglage est annulé par toute autre valeur pour l'option `titlepage`. Les bordures de cette page de couverture sont :

- `\coverpagetopmargin`(bord supérieur),
- `\coverpageleftmargin`(bord gauche),
- `\coverpagerightmargin`(marge droite),

et bien sûr :

- `\coverpagebottommargin`(marge inférieure de la page de garde).

Le bord inférieur est paramétrable. Les réglages sont `\evensidemargin` par défaut en fonction de la longueur et `\topmargin` qui peuvent être modifiés par `\renewcommand`. Dans les classes `scrbook` et `scrreprt`, les pages de titre sont prédéfinies par défaut. En revanche `scrartc` utilise un entête de titre par défaut.

```
\begin{titlepage} ... \end{titlepage}
```

Fondamentalement, dans les classes `standard` et `KOMA-Script`, toutes les pages de titres sont dans un environnement particulier, `titlepage`.

Cette zone commence toujours une nouvelle page de droite recto-verso en mode une colonne. Pour une seule page, le style est modifié avec `\thispagestyleempty`, de sorte que ni page, ni titre courant ne soient attribués. La fin de l'environnement marque la fin de la mise en page. Vous ne pouvez pas utiliser la mise en page automatique du titre, car `\maketitle`, décrit par la suite, offre l'aide pour la conception de cette page.

Exemple : vous voulez une page de titre sur laquelle seule la partie supérieure gauche comprend en grand et gras le mot **Moi** mais aucun auteur, aucune date, ni rien de plus :

```
\documentclass{scrbook}
\begin{document}
  \begin{titlepage}
    \teg{\Huge Moi}
  \end{titlepage}
\end{document}
```

```
\maketitle{numéro de page}
```

Ce code permet d'inclure un argument optionnel utilisé comme numéro de page à partir de la première page de titre. Il affecte la numérotation et vous devez absolument choisir un nombre impair. Deux applications sont utilisées pour l'argument optionnel. Avec la première, vous pouvez donner le numéro logique -1 à la page `demi-titre` afin de démarrer le numéro de la page du titre principal par 1. La page peut commencer par un numéro plus élevé tel que 3, 5 ou 7, de façon à tenir compte d'autres pages de titre qui sont ajoutées par l'éditeur. Si une rubrique est utilisée, l'argument optionnel est ignoré. Par conséquent, le style de ces préliminaires peut être modifié en redéfinissant les macros de `\titlepagestyle`. Les commandes suivantes ne conduisent pas nécessairement à des titres dont la définition est faite par `\maketitle` et il est bon de noter qu'à ce stade, ce dernier ne doit pas être utilisé dans l'environnement `\titlepage`.



Comme indiqué dans les exemples, vous devriez utiliser `\maketitle` ou la page de titre. Les instructions décrites ci-dessous fixent le contenu des titres qui doivent être utilisés nécessairement avant `\maketitle`, alors que lors de l'utilisation de l'ensemble `babel` (voir [Bra01]) il est recommandé de ne pas les inclure dans le préambule avant `\begindocument`.

Des documents exemples figurent à la fin de la section `\enddocument`.

```
\extratitle{demi-titre}
```

Les classes `standard` produisent une page de titre avec les trois éléments titre, auteur et date, `KOMA-Script` peut produire six pages différentes avec `\maketitle`, ce qui rend son utilisation incontournable. Autrefois, un livre était souvent protégé de la pollution par la couverture, tâche qui prenait la première page avec un titre abrégé et même un `demi-titre`. De nos jours, une page supplémentaire réelle est utilisée avant le titre complet et contient des informations sur l'éditeur, le numéro de série ou informations similaires sur le livre. Avec `KOMA-Script`, il est possible de mettre une autre page avant la page de couverture réelle. Le `demi-titre` peut être un texte arbitraire dont le contenu, émis par `KOMA-Script` n'influence pas la mise en forme supplémentaire. L'organisation est complètement laissée à l'utilisateur. Le dos du titre reste vide. La page de `demi-titre` a aussi son propre montage lorsque l'on travaille avec des entêtes de titre et sa sortie, gérée par `\extratitle` a lieu dans le cadre des titres produits par `\maketitle`.

Revenons à l'exemple ci-dessus, et supposons que le spartiate **Moi** est un `demi-titre`. Le titre complet devrait toujours suivre le titre principal. On peut procéder comme suit :

```
\documentclass{scrbook}
\begin{document}
\extratitle{\Huge Moi}
\title{C'est moi}
\maketitle
\end{document}
```

Vous pouvez utiliser le `demi-titre`, le centrer horizontalement et plus :

```
\documentclass{scrbook}
\begin{document}
```

```
\extratitle{\vspace*{4\baselineskip}  
\begin{center}{\Huge Moi}\end{center}}  
\title{C'est moi}  
\maketitle  
\end{document}
```

La commande `\title` est fondamentalement nécessaire lors de la création préliminaire avec `\maketitle` pour que les exemples soient sans erreur.

```
\titlehead{entête}  
\subject{dactylographie}  
\title{titre}  
\subtitle{sous-titre}  
\author{auteur}  
\date{date}  
\publishers{éditeur}  
\and  
\thanks{note}
```

La page du titre principal contient sept éléments et sa sortie se fait dans le cadre des titres produits par `\maketitle`, ainsi que les instructions concernant la définition des éléments correspondants.

Le titre principal est défini avec la commande `\titlehead`, imprimé dans le jeu de bloc normal en haut et sur toute la largeur de la page. Il peut être conçu librement par l'utilisateur. Son écriture est contrôlée et se fait à l'aide des instructions `\setkomafont` et `\addtokomafont`. Le réglage par défaut est indiqué dans la table 2.3 page suivante. Le sujet est indiqué juste au-dessus du titre. Les paramètres de police pour cet élément sont demandés, mais le réglage par défaut, qui figure dans ce tableau, peut utiliser les modes `\setkomafont` et `\addtokomafont` et peut être modifié. Il est affiché dans une très grande police, les paramètres s'appliquent au titre de l'élément qui utilisent la même police (voir la table 2.2 page 45). Les paramètres par défaut utilisent les commandes `\setkomafont` et `\addtokomafont`.

Le titre de l'entête (voir la [section 2.7](#)) peut être modifié, mais la taille n'est cependant pas affectée (voir la table 1.5 page 19). Le sous-titre est affiché juste en dessous du titre dans la police définie par défaut mais peut

TABLE 2.3 – Préréglages de la police pour les éléments du titre

Élément	Par défaut
<code>\author</code>	<code>\Large</code>
<code>\date</code>	<code>\Large</code>
<code>\dedication</code>	<code>\Large</code>
<code>\publishers</code>	<code>\Large</code>
<code>\subject</code>	<code>\normalfont \normalcolor \bfseries\Large</code>
<code>\subtitle</code>	<code>\usekomafont\title\large</code>
<code>\title</code>	<code>\usekomafont\disposition</code>

être modifié en utilisant le mode `\setkomafont` et `\addtokomafont` (voir la figure 2.2 page 45).

Sous le sous-titre suit l’auteur. L’argument `\author` peut en indiquer plusieurs, dont les noms doivent être séparés par `\and`.

L’élément `\author` est prédéfini, mais peut être modifié en utilisant `\setkomafont` et `\addtokomafont`.

Sous le ou les auteurs apparaît la date d’aujourd’hui, `\today` prédéfinie. Utiliser la commande `\date` sans aucune indication pour supprimer cet affichage qui peut être modifié en utilisant le mode contrôle `\setkomafont` et `\addtokomafont` (voir la section 2.7). Le réglage par défaut est indiqué dans la table 2.2 page 45.

Enfin vient l’éditeur. Cette commande peut être également utilisée pour toute autre information de peu d’importance.

La commande `\parbox` est exploitée pour composer l’information sur toute la largeur de la page comme un paragraphe normal, équivalente à l’intitulé de l’entête et que ce champ est mis au-dessus des notes existantes. Les notes sur la page de titre ne sont pas produites par note, mais à l’aide de `\thanks`, elles sont associées aux auteurs avec des symboles utilisés comme marqueur de note au lieu de chiffres. Remarquer que `\thanks` figure à l’intérieur de l’argument d’une autre commande, c’est-à-dire par exemple comme argument de l’auteur de la commande `\author`. Bien que l’impression des composants de titre utilisent des éléments d’égalité

de polices, les valeurs par défaut, figurant dans la table 2.3, peuvent être modifiées à l'aide des commandes `\setkomafont` et `\addtokomafont` (voir section 2.7).

Les renvois ne sont pas des notes de bas de page gérées avec `\footnote` mais créées avec la commande `\thanks` et servent généralement pour les notes associées aux auteurs que l'on marque à l'aide de symboles à la place des numéros. Il est important de noter que `\thanks` est utilisé dans l'argument du reste des instructions, par exemple, `auteur` de la commande `\author`.

TABLE 2.4 – Taille de la police et alignement des éléments de la page de titre principal lors de l'utilisation de `maketitle`

Élément	Instruction	Police	Position
entête	<code>\titlehead</code>	<code>\usekomafont{titlehead}</code>	fixe
Dactylographie	<code>\subject</code>	<code>\usekomafont{subject}</code>	centré
Titre	<code>\title</code>	<code>\usekomafont{title}\huge</code>	centré
Sous-titre	<code>\subtitle</code>	<code>\usekomafont{subtitle}</code>	centré
Auteur	<code>\author</code>	<code>\usekomafont{author}</code>	centré
Date	<code>\date</code>	<code>\usekomafont{date}</code>	centré
Éditeur	<code>\publishers</code>	<code>\usekomafont{publishers}</code>	centré

À l'exception de l'entête et des possibles notes de bas de page, tous les éléments sont centrés horizontalement. L'information est résumée dans la table 2.4.

Noter que `\huge` sera utilisé après commutation de la police de l'élément `\title`, pour le titre principal, et que vous ne pouvez pas en changer la taille de police avec `\setkomafont` ou `\addtokomafont`.

Exemple : vous écrivez une thèse. Il convient de préciser avant que le haut de la page est aligné à gauche et le centre d'études, y compris l'adresse, justifié à droite. Comme d'habitude, une légende est ajoutée, y compris l'auteur et la date d'édition. Le conseiller doit être spécifié, et l'on doit se rendre compte qu'il s'agit d'une thèse, ce qui peut être obtenu comme suit :

---

```

\documentclass{scrbook}
\usepackage[french]{babel}
\begin{document}
\titlehead{{\Large Université de Reims
\hfill SS~2015\\}
Faculté du Temps Passé - Maison Blanche
51100 Reims}
\subject{Thèse}
\title{Simulation d'un anévrisme avec DSP}
\subtitle{Petit mais fin?}
\author{Papi Ray}
\date{30. Juin 2015}
\publishers{Supervisé par le professeur KOMA}
\maketitle
\end{document}

```

Une idée commune mais fautive concerne l'importance de la page de titre principal. On a souvent supposé, à tort, que c'était la couverture du livre ou un cache-poussière et que la première page ne respecte pas les lignes directrices d'une mise en page normale, avec à droite et à gauche des plus grandes marges. Mais si l'on prend un livre que l'on ouvre, on constate que ces pages de titre sont produites par `\maketitle`.

La page de titre, comme le  `demi-titre` , obéit à la même mise en page que l'ensemble du document, telle la couverture ou autre chose que vous auriez créé. Si vous utilisez un format particulier, il n'y a aucune raison de ne pas utiliser un graphisme ou un logiciel de PAO pour vous aider. Un document distinct doit être utilisé pour permettre de le publier plus tard sur un support différent ou avec une autre imprimante.

Depuis KOMA-Script 3.12 on peut voir la première page de titre créée avec `\maketitle` formatée comme une page de couverture.

Dans ce cas, changer de page demande un changement des bordures utilisées (voir option `titlepage=firstiscover`).

<pre> \uppertitleback{<i>retour entête de couverture</i>} \lowertitleback{<i>retour pied de couverture</i>} </pre>
--

La page de titre est laissée vide dans les classes T<sub>E</sub>X standard. Avec KOMA-Script on peut imprimer en double face au dos de la page de titre

principal, cependant une distinction est faite entre deux éléments : `titre d'entête de couverture` et `titre de pied de page`. L'entête peut atteindre le pied et vice versa. Dans l'original en langue allemande de ce manuel, l'avertissement a été mis avec l'aide de `\uppertitleback`.

`\dedication{dédicace}`

KOMA-Script a sa page de dédicace, centrée, dont la police est légèrement plus grande. Un réglage exact pour être mieux vu peut être réalisé par les commandes de modification `\setkomafont` et `\addtokomafont` (voir la section 2.6). Le dos est vide comme la page de retour du `demi-titre`. La page de dédicace est produite par `\maketitle` et doit donc être définie avant cette commande.

Si vous écrivez un livre de poésie consacré à votre femme. Il pourrait ressembler à ceci :

```
\documentclass{scrbook}
\usepackage[french]{babel}
\begin{document}
\extratitle{\teg{\Huge Mon amour}}
\title{À ma Jeanne}
\author{Prince Vaillant}
\date{1429}
\lowertitleback{%
Ce livre de poèmes est écrit à l'aide de
{\KOMAScript} et {\lat} ensemble.
\uppertitleback{%
la même maison d'édition \par
éditions: 1 Exemplaire}
\dedication{%
Pour mon trésor\\
Un amour éternel\\
De ton Prince.
\maketitle
\end{document}
```

Vous pouvez remplacer les marques d'affection en fonction de vos propres préférences.

## 2.8 Le résumé

Notamment avec des articles, plus rarement des rapports, il est imprimé sous le titre et avant la table de contenu. Lors de l'utilisation d'un titre, l'entête du résumé est un bloc en retrait dans la règle, à droite et à gauche. En comparaison, le résumé est plutôt avant le chapitre ou la **section**.

```
abstract=valeur marche/arrêt
```

Dans les classes **standard**, une habitude fréquente permet à **abstract** de centrer le titre résumé avant le résumé. Nous trouvons pratique, à la lecture des journaux, de reconnaître le texte affiché au début d'un article ou d'un rapport. Cela vaut d'autant plus lorsque le résumé vient avant la table des matières et c'est agréable quand ce titre apparaît petit et centré. **KOMA-Script** offre la possibilité, avec l'option **abstract**, de choisir la position ou de désactiver, via le « sommaire », la fonction.

Une valeur de **marche-arrêt** peut être l'une des valeurs standard pour de simples interrupteurs de la table 1.6 page 20, dans **KOMA-Script**, la valeur par défaut est **false**. Pour les livres, un type de résumé différent qui fait souvent partie de l'introduction ou qui est un chapitre séparé à la fin du document, est généralement utilisé. Par conséquent, il n'y a pas d'environnement **scrbook abstract**.

```
\begin{abstract} ... \end{abstract}
```

Certaines classes **L<sup>A</sup>T<sub>E</sub>X** offrent un environnement **abstract** spécial pour ce résumé. Cette sortie directe ne fait pas partie des titres fixés par **\maketitle** et n'est pas un élément de l'ensemble. Rappelez-vous qu'il s'agit d'un environnement, pas d'une commande. Que le résumé soit (ou ne soit pas) fourni avec un titre est contrôlé par l'option **abstract** (voir ci-dessous).

Lorsque vous utilisez la classe **scrreprt**, il est utile d'examiner si vous ne devez pas procéder de même avec les commandes **\chapter\***, **\addchap** ou **\addchap\***. Si une rubrique (voir l'option **titelpage**, section 2.7) est utilisée, le résumé est réglé en interne par le biais d'un environnement **quotation**. Ce paragraphe commence normalement avec le plateau mais ne doit pas être collecté au premier **alinéa**, et il peut, avec **\noindent** être supprimé immédiatement après **\begin{abstract}**.

## 2.9 Table des matières

Les titres existants sont généralement repris dans une table des matières récapitulative et l'on retrouve aussi et souvent, après cette dernière, les listes des flottants, par exemple des tables et des figures (voir section 2.20).

Les fonctions `\DeclareSectionCommand`, `\DeclareNewSectionCommand`, `\RedeclareSectionCommand` et `\ProvideSectionCommand`, peuvent, en plus de ce qui est documenté dans cette section, inclure des options ayant également un impact sur le contenu du répertoire (voir l'article 21.3).

`toc=cadre`

Récemment, il est devenu presque coutumier de prendre les index, tables et figures pour la bibliographie, sans doute en rapport avec le nouveau mode<sup>1</sup> d'insertion de ces répertoires qui présentent une structure et une similarité significative avec la table des matières, à la fin du livre. Cette évolution me rend sceptique. Comme il n'y a pas de sens d'inclure la table des figures sans l'autre table de contenu, ces répertoires s'ajoutent à la table des matières avec le réglage `listof`. Des listes sont également créées qui prennent en compte l'utilisation de flottants, version `paq 1.2e` (voir [Lin01]) ou `floatrow` (voir [Lap08]) pour créer des registres qui citent le contenu d'autres sections de ce travail et pour obtenir des tables ou des figures avec les paqs mentionnés ci-dessus sans numéro de chapitre. Si vous voulez ignorer ce principe, le réglage utilise `listofnumbered`.

L'index d'une entrée est remplacé par `toc=index` dans la table puisque l'index contient également des références aux contenus d'autres sections, y compris cette entrée, et n'a pas de numéro. Une dérogation à ce principe n'est pas supportée par `KOMA-Script` malgré l'intérêt et la prise en charge de `toc` par l'auteur.

La bibliographie est différente avec un autre type de répertoire. Ici, elle se réfère non pas au contenu du présent document, mais plutôt à du contenu externe. On peut presque dire que la bibliographie propre au chapitre est représentative et donc mérite d'être numérotée. L'option `toc=bibliographynumbered` gère justement cet effet, y compris l'entrée tardive dans la table des matières. Avec le même raisonnement, je pense

---

1. il s'agit bien du mode (de la manière) et non de la mode.



qu'une liste classique de sources mérite un chapitre distinct. En outre, la bibliographie n'est rien d'autre que vous n'avez écrit vous-même. Donc l'entrée non numérotée dans la table des matières m'apparaît appropriée, ce que permet le réglage `toc=bibliography`. Normalement, la table des matières est formatée de telle sorte que les niveaux hiérarchiques sont isolés. Une largeur fixe est prévue pour le numéro de division dont le nombre fixé est justifié à gauche, à tous les niveaux. Cela correspond à `toc=graduated`.

S'il existe de très nombreux articles, la hiérarchie des numéros est très large mais l'espace disponible n'est pas suffisant. Dans la FAQ allemande [Wik] il est proposé de redéfinir dans ce cas la table des matières.

KOMA-Script offre un format alternatif avec lequel le problème ne se produit pas. Lorsque vous utilisez l'option `toc=flat`, les différents niveaux de détail ne diffèrent pas beaucoup. Au lieu de cela, une forme de table est choisie dans lequel tous les numéros d'unité et l'ensemble du texte de contour respectivement justifiés à gauche sont dans une colonne. L'espace requis pour les numéros d'unité est déterminé automatiquement.

Ci-après, une vue d'ensemble des valeurs possibles pour le réglage des paramètres de `toc`.

TABLE 2.5 – Valeurs possibles de l'option `toc` pour fixer forme et contenu de la table des matières

Éléments	Description
<code>bibliography, bib</code>	La bibliographie aura une entrée non numérotée dans la table des matières.
<code>bibliographynumbered, numberedbibliography</code>	La bibliographie aura une entrée numérotée dans la table des matières.
<code>bibnumbered, numberedbib</code>	La bibliographie aura une entrée numérotée dans la table des matières.
<code>chapterentrywithdots, chapterentrydotfill</code>	Dans les entrées de chapitre des classes <code>scrbook</code> et <code>scrreprt</code> le texte et le numéro des pages sont reliés par une ligne en pointillés.

<code>chapterentrywithoutdots,</code> <code>chapterentryfill</code>	Dans les entrées de chapitre des classes <code>scrbook</code> et <code>scrreprt</code> le texte et le numéro des pages ne sont pas reliés par une ligne en pointillés.
<code>flat, left</code>	La table des matières est remplacée par une forme de table. Les index de classification sont dans la première colonne, les entêtes dans la seconde et les nombres dans la troisième. L'espace alloué pour le numéro de l'article, dépend lui-même de l'espace requis lors de l'exécution précédente de <code>L<sup>A</sup>T<sub>E</sub>X</code> .
<code>graduated, indent,</code> <code>indented</code>	La table des matières est remplacée par une forme hiérarchique. Il y a seulement un espace limité pour les numéros hiérarchiques.
<code>indenttextentries,</code> <code>indentunnumbered,</code> <code>numberline</code>	La propriété <code>numberline</code> (voir la section 14.2) est réglé pour la table des matières. Ces entrées non numérotées sont alignées à gauche avec le texte d'entrées numérotées qui est défini au même niveau.
<code>index, idx</code>	L'index a une entrée dans la table des matières, sans qu'il soit numéroté.
<code>indexnumbered</code>	L'index a une entrée numérotée dans la table des matières.
<code>leftaligntextentries,</code> <code>leftalignunnumbered,</code> <code>nonumberline</code>	La propriété <code>numberline</code> (voir la section 14.2) supprime la table des matières. Ces entrées non numérotées sont alignées à gauche avec le nombre d'entrées numérotées de même niveau. C'est la valeur par défaut.
<code>listof</code>	La liste des flottants, tels que la cartographie et la liste des tables, obtient une entrée dans la table des matières, sans être numérotée.
<code>listofnumbered,</code> <code>numberedlistof</code>	La liste des flottants, tels que la cartographie et la liste des tables, obtient une entrée dans la table des matières et sont numérotées.
<code>nobibliography, nobib</code>	La bibliographie ne reçoit aucune entrée dans la table des matières.
<code>noindex, noidx</code>	L'index ne reçoit aucune entrée dans la table des matières.

<code>nolistof</code>	Les listes des flottants, tels la cartographie et la liste des tables, ne reçoit aucune entrée dans la table des matières.
<code>sectionentrywithdots,</code> <code>sectionentrydotfill</code>	Dans les entrées de sections, les textes et numéros de page de la classe <code>scrartcl</code> sont reliés par une ligne pointillée.
<code>sectionentrywithoutdots,</code> <code>sectionentryfill</code>	Dans les entrées de sections, les textes et numéros de page de la classe <code>scrartcl</code> ne sont pas reliés par une ligne pointillée.

<code>chapterentrydots=on-off-valeur</code> <code>sectionentrydots=ouvert-fermé-valeur</code>
--

Ces options déterminent si la table des matières dans les classes `scrbook` et `scrreprt` pour le niveau de chapitre ou la classe `scrartcl` pour le niveau de la `section` ou un niveau plus profond utilise une ligne en pointillés reliant le texte de l'article et le numéro de page associé. Quand `on-off-valeur` peuvent être les valeurs standard utilisées pour de simples interrupteurs dans la table 1.6 page 20. La valeur par défaut est `on`.

Les deux options sont fausses si, au lieu d'une ligne en pointillés, une seule distance est utilisée. Si la ligne en pointillés est utilisée, la police des éléments `chapterentrydots` et `sectionentrydots` peut être modifiée au cours des réglages pour les numéros de page dans les entrées (voir la table 2.2 page 45 et la `section` 2.6).

Les paramètres par défaut des éléments sont donnés dans la table 2.6 page 63. Il est à noter que le résultat est correct en utilisant une police identique pour tous les points.

<code>\tableofcontents</code>
-------------------------------

La sortie de la table des matières est effectué par les `\tableofcontents`. Pour obtenir une bonne table des matières, après chaque changement, deux exécutions de `LATEX` sont nécessaires. En utilisant l'option `toc` décrite ci-dessus, la portée et la forme de la table des matières du répertoire concerné sont modifiées. Deux exécutions consécutives de `LATEX` sont nécessaires à l'enregistrement de la modification.

L'entrée pour l'unité de la plus haute `section` ci-dessous `\part`, c'est-à-dire `\chapter` pour `scrbook` et `scrreprt` ou `\section` pour `scrartcl`, ainsi que le niveau de plan `\part` lui-même n'est pas en retrait. Il est, en même temps, situé entre le texte de l'entête de `section` et un numéro de page.

Les raisons typographiques de cet arrangement concernent normalement la police et l'accent désiré. La création originale du contenu de ce manuel est réglée avec les paramètres par défaut et sert d'exemple. L'écriture de ces deux entrées `toc` fait également partie de l'entrée des éléments pour `scrbook` et du chapitre de `scrreprt` ainsi que de l'entrée réglable de `section` pour `scrartcl`.

La police du numéro de la page peut respectivement être réglée pour chacun des éléments `partentrypagenumber` et `chapterentrypagenumber` (`scrbook` et `scrreprt`) ou `sectionentrypagenumber` (`scrartcl`) (voir la `section` 2.6, table 2.2 page 45). Les valeurs par défaut des éléments sont donnés dans la table 2.6 page suivante.

### tocdepth

Normalement, les niveaux hiérarchiques de `\part` jusqu'au paragraphe `\subsection` des classes `scrbook` et `scrreprt` et les mêmes niveaux de la classe `scrartcl` sont inclus dans la table des matières. Ceci est contrôlé par le compteur `tocdepth`, valeur -1 pour `\part`, 0 pour `\chapter` et ainsi de suite. En fixant, augmentant ou diminuant le compteur, les entrées de niveau peuvent être déterminées et apportées à la table des matières, mais elles restent néanmoins des classes `standard`.

Les valeurs de `\tocdepth` et `\secnumdepth` ne correspondent pas à `scrartcl \part`, pour des raisons de compatibilité avec la classe `standard scrartcl` (voir la `section` 2.16). Par conséquent, pour la mise en `tocdepth` la commande `\partnumdepth` doit pas être utilisée.

Vous écrivez un article dans lequel `\subsubsection` est utilisé pour le niveau de contour mais vous ne voulez pas ce niveau hiérarchique dans la table des matières. Le préambule de votre document peut ressembler à :

```
\documentclass{scrartcl}
\setcounter{tocdepth}{2}
```

TABLE 2.6 – Paramètres par défaut de la police pour les éléments de la table des matières

<code>partentry</code>	<code>\usekomafont{disposition}\large</code>
<code>partentrypagenumber</code>	
<code>chapterentry</code>	<code>\usekomafont{disposition}</code>
<code>chapterentrydots</code>	<code>\normalfont</code>
<code>chapterentrypagenumber</code>	
<code>sectionentry</code>	<code>\usekomafont{disposition}</code>
<code>sectionentrydots</code>	<code>\normalfont</code>
<code>sectionentrypagenumber</code>	

Le compteur `tocdepth` est défini sur 2, valeur de `\subsection` et les entrées de `scrartcl` dans la table des matières ont lieu à ce niveau, vous pouvez utiliser cette valeur par défaut mais également la supprimer du compteur `tocdepth` :

```
\documentclass{scrartcl}
\addtocounter{tocdepth}{-1}
```

Pour ajoutez ou soustraire une valeur à `tocdepth`, il vous suffit de compiler la table des matières avec au moins deux exécutions de `LATEX`.

## 2.10 Mise en page des paragraphes

Les classes `standard` fixent normalement les paragraphes avec retrait et sans espace vertical entre les paragraphes. C'est la meilleure solution en utilisant une mise en page régulière, comme celle produite avec le package `typearea`.

Si aucune indentation ni interligne n'est indiquée, seule la longueur de la dernière ligne donnerait au lecteur un point de référence. Dans les cas extrêmes, il est très difficile de détecter si une ligne est pleine ou non. En outre, il est constaté qu'un marqueur à la fin de l'`alinéa` a tendance à être facilement oublié par le début de la ligne suivante.

Un espacement interparagraphe présente l'inconvénient de disparaître dans certains contextes. Un marqueur au début d'un paragraphe est plus facile à retenir car souvent, après l'affichage d'une formule, il est à peu près impossible de détecter si le paragraphe se prolonge ou si c'est une autre **section** qui se met en marche et quand on commence à lire en haut d'une page, il devient indispensable de consulter la page précédente pour voir si un **alinéa** commence ou non. Tous ces problèmes disparaissent en utilisant l'indentation dont la combinaison avec l'espacement vertical entre les paragraphes est redondante et doit être évitée. Son seul inconvénient est la réduction de la longueur de ligne. Mais comme en orthotypographie française tout principe a son exception et celui-ci ne déroge pas à la règle : l'utilisation de lignes courtes, par exemple dans un journal, justifie, en plus de l'indentation, l'espacement entre les paragraphes.

`parskip=méthode`

Il y a parfois des demandes pour une mise en page de document avec un espacement vertical entre les paragraphes au lieu de l'indentation. Les classes **KOMA-Script** fournissent avec l'option `parskip` plusieurs façons d'utiliser, à la place du retrait, l'espacement entre les paragraphes. Le procédé consiste en deux éléments. Le premier est soit **plein** (`full`), soit à moitié plein (`half`), ce qui signifie que la totalité de l'espace d'une ligne ou seulement la moitié d'une ligne est occupé.

La seconde partie concerne les caractères « \* », « + » ou « - », et peut être omise. Sans le deuxième élément, la dernière ligne d'un paragraphe se terminera par un espace blanc d'au moins 1em. Le signe plus laisse libre l'extrémité de la ligne sur au moins un tiers et le signe neutre sur au moins un quart de la ligne. La variante moins ne prend pas soin de l'espace blanc à la fin de la dernière ligne d'un paragraphe. Le réglage peut être modifié de n'importe quel endroit à l'intérieur du document. Dans ce cas, la commande `\selectfont` sera implicitement appelée. Le changement sera valide et vu à partir du paragraphe suivant. Outre les huit combinaisons possibles, les valeurs de simples interrupteurs montrées dans la table 1.6, peuvent être utilisées. L'activation de l'option correspond à l'aide complète sans annexe et se traduit par l'espacement entre les paragraphes avec un blanc de 1em à la fin de la dernière ligne de chaque paragraphe. L'arrêt de l'option réactive par défaut l'espace-tiret de 1em à la première ligne du

paragraphe au lieu de l'espacement des paragraphes. Toutes les valeurs possibles de l'option `parskip` sont présentées ci-après.

TABLE 2.7 – Valeurs possibles de l'option `parskip` pour sélectionner le marquage des paragraphes

Éléments	Description
<code>false</code> , <code>off</code> , <code>no</code>	Les paragraphes sont caractérisés par un retrait de la première ligne d'un blanc de 1em. Le premier paragraphe de la section n'est pas escamoté.
<code>full</code> , <code>true</code> , <code>on</code> , <code>yes</code>	Les paragraphes sont séparés par une distance verticale d'une ligne. Une espace de la police de base d'un blanc d'au moins 1em termine la dernière ligne.
<code>full-</code>	Les paragraphes sont indiqués par une distance verticale d'une ligne. Les extrémités de paragraphe ne sont pas étiquetées.
<code>full+</code>	Les paragraphes sont indiqués par une distance verticale d'une ligne. Les extrémités de paragraphe sont caractérisés par un espace libre d'au moins un tiers d'une ligne normale.
<code>full*</code>	Les paragraphes sont indiqués par une distance verticale d'une ligne. Les extrémités de paragraphe sont caractérisés par un espace libre d'au moins un quart d'une ligne normale.
<code>half</code>	Les paragraphes sont indiqués par une distance verticale d'une demi-ligne. Le paragraphe se termine par une espace libre d'au moins 1em à la fin.
<code>half-</code>	Les paragraphes sont indiqués par une distance verticale d'une demi-ligne. Les extrémités de paragraphe ne sont pas étiquetées.
<code>half+</code>	Les paragraphes sont indiqués par une distance verticale d'une demi-ligne. Les extrémités de paragraphe sont caractérisées par un espace libre d'au moins le tiers d'une ligne normale.
<code>half*</code>	Les paragraphes sont indiqués par une distance verticale d'une ligne. Les extrémités de paragraphe sont caractérisées par un espace libre d'au moins le quart d'une ligne normale.
<code>never</code>	Aucun espace ne sera inséré entre les paragraphes si <code>\flushbottom</code> , distance verticale supplémentaire, est attribuée à l'ajustement de l'équilibre vertical.

Utiliser les environnements d'espacement change la distance avant, après et à l'intérieur des paragraphes de la liste, ce qui évite des problèmes au sein de ces environnements ayant une séparation plus grande que celle entre les paragraphes de texte normal. Matières, listes de figures et tables sont toujours réglées sans espacement des paragraphes supplémentaires. La valeur par défaut de KOMA-Script est `parskip=false`. Dans ce cas, il n'y a pas d'espacement entre les paragraphes, seul un retrait de la première ligne de 1em.

### 2.11 Reconnaissance des pages paire et impaire

Dans les documents recto-verso, une distinction est faite entre les pages de gauche et de droite. La page de gauche porte toujours un numéro pair et celle de droite un numéro impair. Ainsi, dans ce guide, elles sont désignées comme les pages paire et impaire<sup>2</sup>. Cela signifie que la détection et la reconnaissance d'une page gauche ou droite est la même que la détection et la reconnaissance des numéros de pages pair et impair. Ce guide traite de la simplicité de gestion des pages paire et impaire.

Dans des documents recto, la distinction entre les côtés gauche et droit n'existe pas, mais il existe des documents simple face avec des numéros de pages pair ou impair.

```
\ifthispageodd{alors}{sinon}
```

Pour savoir si un texte tombe sur une page paire ou impaire, on utilise la commande `\ifthispageodd`. L'argument `alors` est exécuté uniquement si la commande tombe sur une page impaire. À défaut, l'argument `sinon` est retourné.

Vous pouvez préciser qu'un texte figure sur une page paire ou impaire, à l'aide de l'entrée :

```
ceci est une page avec un numéro  
\ifthispageodd{im}{}pair,
```

dont la réponse pourraient être, par exemple :

```
Ceci est une page avec un numéro de page impair.
```

---

2. PAPIRAY précise : bien entendu, les pages sont plusieurs, la 1 et la 2, par exemple.



Noter que l'argument `else` (sinon) est laissé vide.

La commande `\ifthispageodd` utilise un mécanisme similaire à une étiquette et une référence à elle, ce qui rend une double exécution de  $\text{\LaTeX}$  nécessaire après un changement dans le texte. C'est alors seulement que la décision est correcte. Dans la première passe, une heuristique est utilisée pour la décision. Pour plus d'informations sur le problème de la détection des côtés gauche et droit, ou des numéros de pages paires et impaires consulter la section 20.1 dans le *coin des experts*.

## 2.12 Styles prédéfinis d'entête et pied de page

Une des propriétés générales d'un document est le style de page. Avec  $\text{\LaTeX}$  ce que nous entendons par style concerne en premier lieu le contenu des entête et pied de page.

<code>headsepline=on-off-valeur</code> <code>footsepline=on-off-valeur</code>
--

Ces options sont définies en plaçant, sous les titres courants ou au-dessus du pied de page, une ligne horizontale utilisant une valeur standard `on-off` de la table 1.6 page 20. L'activation de l'option `headsepline` ainsi que son utilisation sans transfert de valeur permet de placer la ligne. Dans les styles de page vide ou recto, cette option n'a pas d'effet car la mise en page ne comporte pas d'entête. L'option `footsepline` dans les mêmes conditions, permet de positionner la ligne au-dessus du pied de page. Sa désactivation arrête les lignes respectives.

Typographiquement on considère que cette ligne donne un effet visuel de rapprochement de l'entête et du texte. Cela ne signifie pas que l'entête doit être éloigné du corps, mais qu'il doit, au contraire, être pris en compte dans le calcul de la zone de type comme appartenant à ce même corps, ce qui est réalisé automatiquement en utilisant l'option de classe `headsepline`.

L'option `headinclude` du paq `typearea` offre la même possibilité. Il en va de même pour `footsepline` et pour `footinclude`.

Contrairement à l'option de `headsepline`, `footsepline` s'applique au style simple, car une page recto est numérotée dans le pied.

Les paqs `scrpage2` et `sclayer-scrpage` offrent plus de possibilités pour les lignes d'influence dans les entête et pied de page.

```
\pagestyle{style de page}
\thispagestyle{style de page locale}
```

Habituellement, une distinction est faite entre les quatre styles différents de page :

- `empty` est le style de page complètement vide pour l'entête et le pied de page qui correspond à la classe `KOMA-Script standard`,
- `headings` est le style de page, avec les titres de colonnes dans lesquels les rubriques sont automatiquement incluses dans l'entête de page. désigné souvent, sur internet ou dans les descriptions de paqs de  $\LaTeX$ , sous le nom anglais « *running headline* ». Dans les classes `scrbook` et `scrreprt` les titres des chapitres et sections dans l'entête sont répétés, avec `KOMA-Script`, dans la mise en page recto-verso sur l'extérieur, et avec les classes `standard`, à l'intérieur.

Avec `KOMA-Script`, le numéro de page est défini dans le pied et dans les classes `standard` à l'extérieur de l'entête. Les titres des chapitres utilisés dans les présentations unilatérales, sont centrés dans l'entête alors que les numéros de page sont centrés dans le pied.

Avec `scrartcl`, le plan respecte la même procédure mais à un niveau plus bas dans la `section` et `sous-section`, le chapitre n'existant pas ici.

Les classes `standard` définissent toujours automatiquement les rubriques en lettres majuscules, mais `KOMA-Script` applique le style trouvé dans le titre. Cela a plusieurs raisons typographiques. Les capitales<sup>3</sup> apparaissent comme une distinction beaucoup trop puissante mais sont malgré tout utilisées. Tout cela n'est pas pris en compte dans les classes `standard`. En outre, les options `headsepline` et `footsepline` de `KOMA-Script` permettent de placer les lignes sous l'entête et sur le pied.

- `myheadings` correspond principalement aux titres `headings` de style de page qui ne sont pas générés automatiquement, mais générés par l'utilisateur en utilisant les commandes `\markboth` et `\markright` ;

---

3. PAPIRAY précise : je ne fais aucune distinction entre « capitale » et « majuscule » que je considère comme synonymes. Pour corroborer ce choix, consulter la note, page 10, de l'ouvrage « Le Ramat européen de la typographie », aux éditions De CHAMPLAIN.

- `plain` est le style de page unique avec l'entête vide. Le numéro de page est délivré. Dans les classes `standard` ce numéro de page est toujours centré dans le pied. Dans `KOMA-Script` la sortie se fait au format double face à la place à l'extérieur dans le pied. Le style de page à un seul côté correspond à celui de la classe `KOMA-Script standard`.

Utiliser la commande `\pagestyle` permet de régler ce style de page. Avec `\pagestyle` avant l'instruction de saut de page, le changement s'applique à la nouvelle page créée après l'utilisation de la fonction `\cleardoublepage`.

Utiliser à la place, la commande `\thispagestyle` pour un changement du style de la page en cours, ce qui arrive aussi automatiquement dans le document, par exemple, dans la déclaration implicite des premières pages de chapitres, exécuter `\thispagestyle{\chapterpagestyle}`

TABLE 2.8 – Paramètres par défaut de la police des éléments style de page

Élément	Par défaut
<code>pagefoot</code>	
<code>pageheadfoot</code>	<code>normalfont normalcolor slshape</code>
<code>pagenumber</code>	<code>normalfont normalcolor</code>

Noter que le changement entre les rubriques automatiques et manuelles pour l'utilisation des paqs `scrpage2` ou `sclayer-scrpage` ne s'opère plus sur le style, mais à l'aide d'instructions spécifiques. Les deux styles de page `headings` et `myheadings` ne doivent pas être utilisés avec ces paqs (voir également chapitre 5).

Pour changer la police de l'entête et du pied de page ou le numéro de page, utilisez les commandes `\setkomafont` et `\addtokomafont` (voir section 2.7). Le même élément `pageheadfoot` est responsable de l'entête et du pied. L'élément pour le numéro de page dans l'entête ou le pied est nommé `pagenumber`. `pagefoot` est également fourni et utilisé dans les classes élément `KOMA-Script` si vous avez défini un style de page avec les paqs `sclayer-scrpage` ou `scrpage2`, qui contiennent également le texte de pied. Les paramètres par défaut peuvent être trouvés dans la table 2.5 page 59. Vous voulez mettre entête et pied de page dans une taille plus petite et en italique, mais souhaitez que le numéro de page ne change pas. Même si le résultat est laid, vous pouvez réaliser ce qui suit :

```
\setkomafont{pageheadfoot}\%  
\normalfont\normalcolor\itshape\small  
\setkomafont{pagenumber}{\normalfont\bfseries}
```

Voulez-vous que, en plus de la variante déjà prédéfinie, une taille de police plus petite soit utilisée :

```
\addtokomafont{pagehead}{\small}
```

Comme vous pouvez le voir, l'élément a utilisé `pagehead` dans le dernier exemple. Vous obtiendrez le même résultat si vous utilisez `pagehead` dans le pied (voir la table 2.5 page 59).

Il n'est pas possible de forcer automatiquement les capitales pour les titres courants, à ce stade, mais si vous le voulez, vous devez redéfinir `\MakeMarkcase` en conséquences et utiliser le paq `scrlayer-scrpage`.

Si vous voulez définir votre propre style pour la page, les commandes :

- `\usekomafont{pageheadfoot}`
- `\usekomafont{pagenumber}`
- `\usekomafont{pagefoot}`

pourraient vous être utiles.

Si vous n'êtes pas sûr de l'ensemble `scrlayer-scrpage` KOMA-Script ou `scrpage2`, alors choisissez, par exemple, la `fancyhdr` dont vous pouvez utiliser les commandes dans vos définitions.

Les effets `fncychap` sont compatibles KOMA-Script et s'utilisent pour la mise en page dans vos propres définitions. Les paqs `scrlayer-scrpage` et `scrpage2` permettent une compatibilité maximale.

```
\markboth{repère gauche}{repère droite}  
\markright{repère droite}
```

Lorsque, dans les rubriques du style de page `myheadings`, les titres des colonnes ne sont pas réglés automatiquement, j'utilise les commandes `\markright` et `\markboth`. Le repère de gauche est habituellement côté gauche de l'entête et le repère de droite marque le côté droit. Dans le cas d'un seul côté n'existe que la marque de droite. La commande `\markleft` est utilisée par les paqs `scrlayer-scrpage` ou `scrpage2`.

Les instructions peuvent également être utilisées avec d'autres modèles. Combiné avec rubriques automatiques ou style de rubriques, l'effet est limité jusqu'au prochain réglage automatique du choix correspondant.

<pre> \titlepagestyle \partpagestyle \chapterpagestyle \indexpagestyle </pre>
---

Sur certaines pages, un style différent est sélectionné automatiquement en utilisant `\thispagestyle`.

TABLE 2.9 – Macros pour définir le style de pages spéciales

<code>\titlepagestyle</code>	page avec les titres entête de titre (voir <a href="#">section 2.7</a> )
<code>\partpagestyle</code>	page pour les pages avec des parties titrées <code>scrbook</code> (voir <a href="#">section 2.16</a> )
<code>\chapterpagestyle</code>	sur les premières pages de chapitres <code>scrbook</code> (voir <a href="#">section 2.16</a> )
<code>\indexpagestyle</code>	style de la première page du répertoire des mots clés (voir <a href="#">section 2.24</a> )

Deux des macros `\partpagestyle` et `\chapterpagestyle` sont utilisées seulement pour les classes `scrbook` et `scrartcl`. Par défaut, le style de page dans les quatre cas est clair et peut être redéfini en utilisant `\renewcommand`.

Exemple si vous ne voulez pas que les pages soient numérotées, utiliser la commande suivante dans le préambule de votre document :

```
\renewcommand*{\partpagestyle}{empty}
```

Comme vous l'avez vu précédemment, le style de page vide est ce qui est justement nécessaire dans cet exemple. Bien sûr, vous pouvez également utiliser un style de page défini par l'utilisateur. Supposons que vous avez défini, avec le paq `scrlayer-scrpage`, votre propre style de page, auquel vous avez donné un nom de style approprié. Pour l'utiliser, il vous faudra définir `\chapterpagestyle` selon :

```
\renewcommand*{\chapterpagestyle}{nom du chapitre}
```

Vous ne voulez pas, à la création, d'une table totalement paginée avec les matières d'un livre et par la suite, que les rubriques de style de page fonctionnent à nouveau. Vous pouvez, par exemple, utiliser :

```
\clearpage
\pagestyle{empty}
\renewcommand*{\chapterpagestyle}{empty}
\tableofcontents
\clearpage
\pagestyle{rubrique}
\renewcommand*{\chapterpagestyle}{plain}
```

Vous pouvez de cette manière conserver la redéfinition locale du style de page pour les pages initiales de chapitre. Ceci a l'avantage de n'avoir à faire aucune hypothèse sur le réglage valide avant le changement que vous pourrez effectuer localement :

```
\clearpage
\begingroup
  \pagestyle{empty}
  \renewcommand*{\chapterpagestyle}{empty}
  \tableofcontents
  \clearpage
\endgroup
```

Il n'est pas recommandé de numéroter une `section` dans un groupe. Cela peut, avec des commandes comme `\label`, conduire à des résultats imprévisibles. Si vous pensez pouvoir également travailler sur les pages de chapitre initial avec des titres en cours d'exécution, simplement avec :

```
\renewcommand*{\chapterpagestyle}{rubriques}
```

consulter `\rightmark`, section 20.1 pour plus d'informations.

```
\pagenumbering{style de numérotation}
```

Cette déclaration travaille avec KOMA-Script de la même manière qu'avec les classes `standard` en utilisant une déclaration du noyau L<sup>A</sup>T<sub>E</sub>X pour faire basculer le style de numérotation pour les numéros de pages.

TABLE 2.10 – Styles de numérotation disponible pour numéros de page.

style de numérotation	Exemple	Importance
<code>arabic</code>	8	chiffres arabes
<code>roman</code>	viii	chiffres romains minuscules
<code>Roman</code>	VIII	chiffres romains majuscules
<code>alph</code>	h	minuscules
<code>Alph</code>	H	majuscules

Les changements prennent effet immédiatement sur la page d’appel de la déclaration et peuvent être mis en place plus vite avec `\cleardoublepage` ou `\clearpage` et les valeurs possibles de numérotation qui figurent dans la table 2.10. L’appel à `\pagenumbering` réinitialise le numéro de page. La page en cours porte le numéro 1 dans le style de numérotation choisi.

## 2.13 Pages vierges

Des pages blanches sont les pages fixées à un document en  $\text{\LaTeX}$ , qui restent en blanc intentionnellement mais sont définies par défaut pour le style de la page en cours de validité. `KOMA-Script` propose ici diverses extensions. Des pages blanches se trouvent généralement dans les livres ayant des chapitres qui commencent côté droit, et dans le cas où le chapitre précédent se termine lui aussi, sur un côté droit, un vide sera inséré du côté gauche. Cette mise en page résulte du fait que les pages blanches existent par défaut dans la présentation double face.

```
cleardoublepage=style de page
cleardoublepage=courant
```

Avec cette option, le style de page blanche insérée par `\cleardoublepage` peut être défini et utilisé comme style de côté (voir la section 2.12).

En outre, `cleardoublepage=courant` est possible, cela correspond par défaut à la version 2.98c de `KOMA-Script` et provoque la création d’une page vide avec le style en cours lors de l’insertion. Depuis la version 3.00 les pages vierges sont créés avec le style de page vide dans un réglage selon les

conventions typographiques de la langue sélectionnée, si vous n'avez pas réglé la compatibilité aux versions antérieures de KOMA-Script. Si vous voulez que les pages vierges à l'exception de la pagination soient vides et générées par le style de page `plain`, utiliser :

```
\KOMAOptions{cleardoublepage=plain}
```

Voir plus de détails sur le style de page dans la section 2.12. Le noyau de L<sup>A</sup>T<sub>E</sub>X intègre la commande `\clearpage`, qui vérifie la publication de tous les flottants, puis commence une nouvelle page.

La commande `\cleardoubleoddplainpage` modifie le style de page vide à gauche de l'attribut `plain` et supprime le titre de la colonne. Dans la mise en page recto-verso, la commande `\cleardoublepage` agit comme `\clearpage` (voir l'option `twoside` dans la section 2.4), crée une page de droite nouvelle, la page gauche est vidée si cela est nécessaire dans le style actuel. Dans KOMA-Script, `\cleardoubleoddstandardpage` agit de la même manière que celle décrite pour les classes `standard`.

```
\clearpage
\cleardoublepage
\cleardoublepageusingstyle{style de page}
\cleardoubleemptypage
\cleardoubleplainpage
\cleardoublestandardpage
\cleardoubleoddpge
\cleardoubleoddpgeusingstyle{style de page}
\cleardoubleoddemptypage
\cleardoubleoddplainpage
\cleardoubleoddstandardpage
\cleardoubleevenpage
\cleardoubleevenpageusingstyle{style de page}
\cleardoubleevenemptypage
\cleardoubleevenplainpage
\cleardoubleevenstandardpage
```

De manière analogue, la commande `\cleardoubleoddemptypage` ôte à la fois le titre et le numéro de page dans le style `empty` de la page de gauche en cours d'exécution. Le site est maintenant vide. Vous pouvez spécifier un style de page personnalisé pour le côté vide, en le prenant



comme argument de `\cleardoubleoddpageusingstyle`. Ici, le style de la page qui sera utilisé est déjà défini.

Parfois, vous ne voulez pas que les chapitres commencent par une page nouvelle, à droite, mais sur une double page à gauche. Cette entorse à la typographie classique peut devenir un privilège si la double page au début du chapitre a un contenu plus spécifique. C'est pourquoi la commande standard `\cleardoubleevenstandardpage` est analogue à la commande `\cleardoubleoddstandardpage`, mais la page suivante est une page de gauche.

Même application en utilisant le style de page `plain` pour les commandes `\cleardoubleevenplainpage` `\cleardoubleevenemptypage` de page vide et `\cleardoubleevenpageusingstyle`.

Le fonctionnement des commandes :

- `\cleardoublestandardpage`,
- `\cleardoubleemptypage`,
- `\cleardoubleplainpage`

et l'argument de la commande :

- `\cleardoubleoddpageusingstyle`

ainsi que la commande :

- `\cleardoublepage`

de l'option précédemment déclarée agissent en fonction du paramétrage expliqué dans les instructions des paragraphes précédents.

Si par exemple vous souhaitez mettre une double page à l'intérieur d'un simple document où bien commencer un nouveau chapitre sur le côté gauche. Si le chapitre précédent se termine avec un côté gauche, une page complètement vide doit être insérée et le côté gauche n'aura ni entête, ni pied de page.

```
\KOMAOPTIONS {cleardoublepage=\index{style de page!
cleardoublepage=!empty}empty}
```

vérifie que les pages vides avec le style de page `\empty` sont sans entête ni pied de page.

Ce réglage peut être déjà fait dans le préambule du document et les options spécifiées comme arguments de `\documentclass`.

Vous écrivez à l'endroit désiré dans le document :

```
\cleardoubleevenemptypage
\thispagestyle{empty}
\includegraphics[width=\textwidth,%
height=\textheight,%
keepaspectratio]%
{figure}
\chapter {entête\index{style de page!entête} de chapitre}
```

La première ligne change le côté à gauche et insère à cette fin si nécessaire un côté complètement vide à droite. La deuxième ligne assure que le côté gauche est également mis à la page de style vide. Les instructions de la troisième à la sixième ligne chargent le fichier de figures avec le nom de la figure et l'ajustent à la taille désirée sans la déformer. Pour cela, l'extension `graphicx` est nécessaire (voir [Car99d]). La dernière ligne commence la page d'un nouveau chapitre, à droite.

## 2.14 Notes de pied de page

Contrairement à l'écriture des classes, KOMA-Script offre la possibilité de configurer les notes de pied de page.

`footnotes=disposition`

Les notes sont marquées dans le texte par défaut. Si un passage comporte plusieurs notes consécutives, il peut donner l'impression que ce ne sont pas deux notes individuelles, mais une seule note avec un nombre élevé. Avec `footnotes=multiple` les notes se succèdent directement, bordées par des séparateurs. Le caractère séparateur défini dans `\multfootsep` est une virgule. L'ensemble du mécanisme est compatible avec `footmisc`. La version 5.3d (voir [Fai11]) mise en œuvre affecte à la fois la marque de référence créée par `\footnote`, et celles émises par `\footnotemark`.

Il est possible de revenir, à tout moment, avec `footnotes=nomultiple` dans `\KOMAoptions` ou `\KOMAoption` aux notes de bas par défaut. D'autres paqs qui influent sur les notes de cette option ne doivent pas être utilisés.

TABLE 2.11 – Valeurs possibles de l’option `footnote`

<code>multiple</code>	les marqueurs consécutifs de bas de page sont séparés par la sortie <code>\multifootsep</code> .
<code>nomultiple</code>	les marqueurs de bas de page immédiatement consécutives sont également émis dans une séquence continue

```

\footnote{numéro}{texte}
\footnotemark{numéro}
\footnotetext{numéro}{texte}
\multiplefootnoteseparator
\multifootsep

```

Les notes sont générées dans KOMA-Script avec `\footnotetext` comme avec `\footnotemark` et `\footnote` dans les classes `standard`. Une note peut être déplacée sur une nouvelle page ce qui se produit généralement lorsque la marque de référence associée est placée en bas de page et qu’il n’existe pas d’autre choix. Au contraire des classes `standard`, KOMA-Script offre la possibilité de détecter automatiquement les notes qui se suivent immédiatement et de les écarter par un caractère de séparation. Pour mettre en place ce séparateur manuellement, il vous suffit d’appeler la commande `\multiplefootnoteseparator`. Toutefois, cette déclaration ne devrait pas redéfinir l’utilisateur ni la mise en forme du séparateur dans le choix, par exemple, de la taille des caractères ou de l’indice supérieur utilisé. Le séparateur lui-même est stocké par défaut dans la commande `\multifootsep` :

```
\newcommand*{\multifootsep}{,}
```

Cela peut être redéfini. Ainsi, pour inclure deux notes, vous écrivez :

```
mot\footnote{note 1}\footnote{note 2}.
```

Les notes sont numérotées 1 et 2. Les deux numéros de bas de page se suivent et donnent l’impression que le mot a une seule note qui porte le numéro 12. Vous pouvez le modifier maintenant avec :

```
\KOMAoptions{footnotes=multiple}
```

pour activer la détection automatique des groupes de note.

Vous pouvez aussi utiliser :

```
mot\footnote{note 1}%  
\multiplefootnoteseparator  
\footnote{note 2}
```

Ce système fonctionne même si la détection automatique échoue ou ne peut pas être utilisée pour une raison quelconque.

Vous souhaitez les numéros de bas de page suivis non seulement par une virgule, mais aussi par une espace pour séparer ? Dans ce cas, écrivez :

```
\renewcommand*{\multfootsep}{\nobreakspace}
```

En préambule du document `\nobreakspace` est choisi ici en place d'un signe de l'espace normale, et ne figure pas dans le classement de la marque de référence :

```
\footref{référence}
```

Parfois, vous obtenez une note de bas dans le texte qui donne plusieurs références dans un document.

La mauvaise solution serait d'utiliser `\footnotemark` indiquant le nombre souhaité. La pire chose à propos de cette solution est que vous devez indiquer un nombre qui peut changer à tout moment. Par conséquent KOMA-Script propose d'utiliser le mécanisme `\label` pour référencer les notes.

L'instruction qui figure sur l'étiquette de `\footref` peut être intégrée par tous les marquages.

Si vous écrivez un texte dans lequel il est nécessaire d'ajouter, à chaque nom, une note indiquant que ce nom est celui d'une marque protégée :

```
La société Splisch \footnote {Ce nom est marque déposée, avec  
droits réservés au propriétaire. \label{refnote}} fournit la  
prochaine SplischPups \footref{refnote} dont il existe la ver-  
sion améliorée Splatsch \footref{refnote}.
```

La note est mise trois fois, une fois directement avec `\footnote` et deux fois avec `\footref`.

Comme les marques de bas de page peuvent être définies en utilisant `\label` pour les changements qui affectent la numérotation des notes,

deux exécutions de L<sup>A</sup>T<sub>E</sub>X seront nécessaires pour enregistrer la nouvelle valeur de `\footref`.

```
\deffootnote[largeur de marque]{collection} {retrait de
paragraphe}{définition de la marque}
\deffootnotemark{définition de la marque}
\thefootnotemark
```

Les classes KOMA-Script ne gèrent pas les notes différemment des classes standard. La marque de référence de la note dans le texte est la même (avec des petits chiffres en exposant) que dans les classes standard et la même mise en forme est reproduite dans la note elle-même. Vous alignez à droite la largeur de la marque et la première ligne de la note suit sur la boîte de la vérification.

Toutes les autres lignes sont émises en retrait et si l'option de largeur n'est pas spécifiée, elle correspond à la valeur de chargement. Pour une note constituée de plusieurs points, la première ligne du paragraphe est prévue avec une indentation.

La figure page suivante illustre à nouveau les différents paramètres.

Dans les classes KOMA-Script, le réglage va avec la définition suivante :

```
\deffootnote[1em]{1.5em}{1em}{%
\textsuperscript{\thefootnotemark}%
}
```

La position haute est atteinte en utilisant `\textsuperscript` avec la plus petite police. La commande `\thefootnotemark` renvoie une marque de référence actuelle sans aucun formatage. Dans la note, avec le marquage se trouve aussi l'élément qui demande la sélection d'une police.

Pour modifier l'étiquette, la police peut utiliser les modes `\setkomafont` ou `\addtokomafont` (voir la table 2.2 page 45 et la section 2.6), réglage par défaut sans changement de police, l'indice supérieur et la taille de police plus petite sont obtenus à l'aide de `textsuperscript`.

L'expression `thefootnotemark` retourne la marque actuelle de la note sans aucun formatage. Le paq `scrextend`, cependant, laisse dans le réglage par défaut des notes de la classe utilisée.

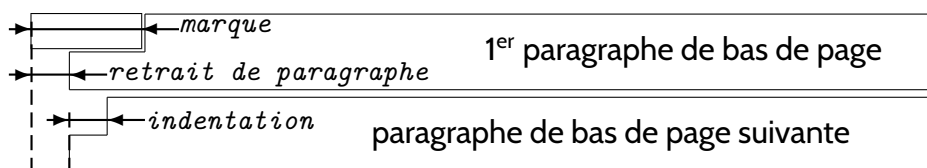


FIGURE 2.1 – Paramètres de la présentation des notes

Le chargement du paq seul doit encore donner lieu à des changements dans la mise en forme des notes ou des marques de bas de page. Plutôt qu’accepter les paramètres par défaut des classes KOMA-Script on peut réaliser soi-même le réglage. A cet effet, la ligne de code ci-dessus peut par exemple être insérée immédiatement après le chargement de `scrextend`.

Dans la note de pied de page se trouve aussi la note incluant la police pour l’élément sélectionné. Cependant, la police de caractères de l’étiquette peut être réglée différemment à l’aide des commandes `\setkomafont` et `\addtokomafont` (voir section 9.6).

Voir aussi la liste des éléments dont la poice peut être modifiée dans la table 3.2 page 176.

Cependant, les éléments trouvés dans `scrextend` s’appliquent lorsque les notes sont établies avec ce paq, c’est-à-dire que `\deffootnote` a été utilisé.

Merci de ne pas abuser de cet élément à d’autres fins, par exemple pour l’utilisation de justification dans les notes (voir aussi `\raggedfootnote`).

La marque de référence dans le texte est définie séparément du marqueur avant la note avec la commande `\deffootnotemark`, réglée par défaut :

```
\deffootnotemark{\textsuperscript{\thefootnotemark}}
```

La police résulte de l’application de `footnotereference` (voir la table 2.2 page 45) avec un repérage identique dans le texte et dans la note. Elle peut être changée avec les commandes `\setkomafont` et `\addtokomafont` (voir la section 2.6).

Une caractéristique souvent souhaitée concerne les marques de bas de page, ni en exposant, ni en plus petit, qui ne doivent pas toucher le texte mais être légèrement en avant de celui-ci, ce qui peut être réalisé avec :

```
\deffootnote{1em}{1em}{\thefootnotemark\}
```

La marque de référence et l'espace qui suit sont donc alignées à droite, dans une boîte de largeur 1em et les lignes de texte de la note sont également en retrait de 1em sur le bord gauche suivant. Une autre disposition souvent demandée concerne des labels alignés à gauche dans la note, ce qui peut être obtenu avec la définition suivante :

```
\deffootnote{1.5em}{1em}{
\makebox[1.5em][l]{\thefootnotemark}%
}
```

Toutefois, si les notes sont dans une police qui doit être transformée, par exemple, en police sans empattement, il suffit d'utiliser `\setkomafont` et `\addtokomafont` (voir la section 2.6) :

```
\setkomafont{footnote}{\sffamily}
```

Comme le montrent les exemples, l'interface KOMA-Script permet à un utilisateur une large variété de formatage de bas de page.

```
\setfootnoterule{hauteur}{longueur}
```

Habituellement, une ligne de séparation se trouve entre la zone de texte et les notes de bas, mais en général elle n'occupe pas toute la largeur de la zone de saisie. Grâce à cette commande vous pourrez déterminer sa longueur exacte, sa hauteur ou son épaisseur. Ici, la hauteur et la longueur ne sont pas évaluées selon le réglage de la ligne lors de sa création avec `\normalsize`. Le niveau de paramétrage optionnel peut être non fixé et ne sera pas modifié, de même que si l'argument hauteur ou longueur est vide, la taille respective n'est pas modifiée et un avertissement sera émis lors de l'utilisation de tailles avec des valeurs non plausibles. La couleur de la ligne peut être réglée à l'aide de l'élément `footnoterule` en utilisant les instructions `\setkomafont` et `\addtokomafont` (voir la section 2.6). Pour changer la couleur, un paq de couleur, comme `xcolor`, doit être chargé.

```
\raggedfootnote
```

Par défaut, les notes dans KOMA-Script sont définies et justifiées comme dans les classes standard mais il est également possible de modifier le reste du document dans un format différent en utilisant `\raggedfootnote` à cet effet mais aussi des définitions valables de `\raggedright`, `\raggedleft`, `\centering`, `\relax` ou selon le réglage par défaut une définition vide. En

outre, les commandes d'orientation du paq `ragged2e` sont autorisées (voir [Sch09]).

Supposons que vous utilisez exclusivement des notes de pied de page pour indiquer à l'évidence des liens très longs conduisant à une rupture justifiant des résultats médiocres, vous pouvez facilement passer avec

```
\let\raggedfootnote\raggedright}
```

les notes de bas de page justifiées à gauche dans le préambule du document.

## 2.15 Délimitations

Dans un livre, une séparation existe entre l'entête, le corps et la suite, possibilité que KOMA-Script intègre pour `scrbook`.

<code>\frontmatter</code>
<code>\mainmatter</code>
<code>\backmatter</code>

Avec `\frontmatter` les suppléments, numérotés en chiffres romains sont introduits. Les titres des chapitres ne sont pas numérotés dans le générique d'ouverture. Les titres des sections ont été numérotés mais sont sortis du chapitre numéro 0 ce qui n'est pas grave, parce que les compléments sont utilisés au maximum pour la matière première, le contenu, les tables de figures, les listes des tables et une préface. L'avant-propos peut être défini comme un chapitre normal, court, et ne doit jamais être divisé en sections. Par conséquent, il n'est pas nécessaire à ce niveau d'ajouter plus de détails.

Dans le cas où l'utilisateur préfère un aspect différent et veut sections et chapitres avec suppléments, les articles de la version 2.97e ne contiennent pas de numéro de chapitre. Ce changement ne vaut que si une compatibilité avec la version 2.97e est paramétrée (voir la version de l'option `section` 2.2). Il est précisé que la confusion des nombres se fait avec les numéros de chapitre. L'utilisation de `\addsec` et `\section*` (voir la `section` 2.16) est du point de vue de l'auteur fortement conseillée et figure dans le générique d'ouverture.



TABLE 2.12 – Valeurs possibles de l’option pour sélectionner les intercalaires avec des pages blanches `scrbook` et `scrreprt`


---

<b>any</b>
Les commandes <code>cleardoublestandardpage</code> , <code>cleardoubleplainpage</code> , <code>cleardoublepageusingstyle</code> , <code>cleardoubleemptypage</code> et la commande <code>cleardoublepage</code> génèrent un saut de page qui correspond à celui de <code>clearpage</code> .
<b>left</b>
Les commandes <code>cleardoublepageusingstyle</code> , <code>cleardoubleemptypage</code> , <code>cleardoubleplainpage</code> , <code>cleardoublepagestandardpage</code> ainsi que la commande <code>cleardoublepage</code> génèrent, quand c’est nécessaire, un saut de page puis insèrent un vide sur le côté à gauche dans la disposition à double face.
<b>right</b>
<code>cleardoublepage</code> , <code>cleardoubleemptypage</code> , <code>cleardoublepageusingstyle</code> , <code>cleardoublepagestandardpage</code> et <code>cleardoubleplainpage</code> génèrent quand c’est nécessaire un saut de page puis insèrent sur le côté droit un vide dans la disposition à double face.

---

Depuis la version 2.97e, les chiffres pour les environnements flottants tels que tables, illustrations et numéros assimilés dans l’entête ne contiennent aucune partie du numéro de chapitre. Cela nécessite un paramètre de compatibilité approprié (voir la version de l’option, [section 2.2](#)).

La partie principale est lancée avec `\mainmatter`. Sans supplément, cette déclaration peut être omise. Dans le corps, les pages sont numérotées en chiffres arabes qui commencent avec le chiffre 1.

L’appendice est intégré à l’aide de `\backmatter`. Parfois on ne trouve dans l’annexe que l’index et d’autre fois la bibliographie. Parfois, l’annexe apparaît et utilise des classements similaires aux positions hiérarchiques. Une numérotation des pages séparées n’est pas fournie et si vous en avez besoin, utiliser la commande `\pagenumbering` de la [section 2.12](#).

## 2.16 Structure du document

La structure se réfère à la classification d'un document en parties, en chapitres, en sections et autres niveaux.

`open=méthode`

Dans les classes KOMA-Script `scrbook` et `scrreprt`, on peut choisir où commence un nouveau chapitre avec une présentation double face. Par défaut, `scrreprt` commence un nouveau chapitre sur la prochaine nouvelle page, alors que `scrbook` commence un nouveau chapitre sur une nouvelle page de droite, méthode commune dans la plupart des livres. Il arrive cependant que, dans certains cas, un nouveau chapitre commence sur une double page, du côté gauche. Un résumé des valeurs possibles se trouve dans la table 2.12 page précédente.

En plus des débuts de chapitre, l'option agit également sur l'utilisation de la commande `\cleardoublepage`, sur l'utilisation des commandes `\cleardoublepageusingstyle`, `\cleardoublestandardpage` ainsi que les commandes `\cleardoubleplainpage` et `\cleardoubleemptypage`. Pour plus de détails, reportez-vous à la section 2.12. L'option ne fait pas de distinction provoquant un effet unilatéral entre les côtés gauche et droit. Dans la classe `scrartcl` le niveau hiérarchique le plus élevé se trouve en dessous de la partie de la section. Par conséquent `scrartcl` ne prend pas en charge l'option.

`chapterprefix=valeur de marche-arrêt`  
`appendixprefix=valeur de marche-arrêt`  
`\IfChapterUsesPrefixLine{alors}{sinon}`

Dans les classes standard de book et report, les entêtes de chapitre sont émises sous la forme d'une rangée chapitre<sup>4</sup> dont le premier numéro est 1, jusqu'à la sortie de la ligne suivante alignée à gauche. KOMA-Script permet ce comportement avec l'option de classe `chapterprefix` avec valeur de `on-off` qui peut être l'une des valeurs par défaut utilisée pour de simples interrupteurs de la table 1.6 page 20.

---

4. Si vous utilisez une langue autre que le français, « chapitre » est bien sûr mis dans la langue correspondante.

La valeur par défaut est `chapterprefix=false`, mais le comportement des classes `standard` correspond à `chapterprefix=true`, options gênant le réglage automatique du titre de chapitre (voir section 2.12) mais vous pouvez mettre les entêtes de chapitre de la partie principale sous la forme simple de `chapterprefix=false`.

Dans le même temps, les rubriques attachées doivent être suivies d'un préfixe « Annexe » avec la lettre de l'annexe prévue, conduisant finalement à l'option `chapterprefix` qui sera changée automatiquement au début de l'annexe. La commande `\IfChapterUsesPrefixLine` exécute du code en fonction de la valeur actuelle de la ligne de préfixe. Si `chapterprefix` est actif, la commande est exécutée, sinon ce sera la partie `else` (sinon) par défaut.

Ce réglage est théoriquement possible avec `appendixprefix=true` mais l'usage en a montré les limites de compatibilité et il est préférable de ne pas l'utiliser.

La police de la ligne de numéro de chapitre de `chapterprefix=true` ou de `appendixprefix=true` peut être modifiée avec les deux instructions `\setkomafont` et `\addtokomafont` (voir section 2.6) sinon un paramétrage par défaut est utilisé, voir la table 2.15 page 95.

`headings=cadre`

Les titres sont normalement fixés assez grands dans les deux classes `standard`, ce qui peut gêner lors de l'utilisation de papier petits formats. Par conséquent, en plus de l'option `headings=big`, deux possibilités peuvent être prédéfinies : `headings=normal` et `headings=small`. Pour les tailles des options de police des titres des classes `scrbook` et `scrreprt`, reportez-vous table 2.15 page 95. Les petites tailles de police sont généralement utilisées avec `scrartcl`. L'espacement est aussi influencé par ces options avant et après les titres de chapitre.

Les options `headings=twolinechapter` et `headings=onelinechapter` figurent dans les titres de chapitres affectant `chapterprefix=true` et `chapterprefix=false` auxquelles correspondent les options des annexes `headings=twolineappendix` et `headings=onelineappendix` ainsi que les annexes `appendixprefix=true` et `appendixprefix=false` (voir ci-dessus).

Les options `headings=openany`, `headings=openright` ainsi que l'option `headings=openleft` pour débiter un chapitre, sont des alternatives à

l'utilisation de l'option ouverte réglée avec les valeurs de `any`, `right` et `left` (voir ci-dessus).

Une autre particularité concerne le traitement de l'argument option des commandes de sectionnement, la fonction et l'importance de ce qui peut être fait par `heading=optiontohead`, `heading=optiontotoc` ainsi que par `heading=optiontoheadandtoc` (un résumé des paramètres possibles pour l'option `headings` dans la table 2.9 page 71). Des exemples de l'utilisation de certains des paramètres possibles sont inclus dans les descriptions des commandes suivantes de post-structure.

TABLE 2.13 – Valeurs possibles de l'option rubriques pour régler les entêtes

Éléments	Description
<code>big</code>	Utiliser des rubriques avec de grands intervalles au-dessus et au-dessous.
<code>normal</code>	Utiliser des titres avec des intervalles moyens au-dessus et au-dessous.
<code>onelineappendix</code> , <code>noappendixprefix</code> , <code>appendixwithoutprefix</code> , <code>appendixwithoutprefixline</code>	Le titre des chapitres dans l'annexe est géré comme les autres rubriques.
<code>onelinechapter</code> , <code>nochapterprefix</code> , <code>chapterwithoutprefix</code> , <code>chapterwithoutprefixline</code>	Le titre des chapitres est prévu comme pour les autres rubriques.
<code>openany</code>	L'instruction <code>\cleardoublepageusingstyle</code> , <code>\cleardoublestandardpage</code> , <code>\cleardoubleplainpage</code> , <code>\cleardoubleemptypage</code> et <code>\cleardoublepage</code> correspond respectivement à une rubrique de <code>headings=openright</code> . Pour les pièces, chapitres, index et annexes, utiliser <code>\clearpage</code> plutôt que <code>\cleardoublepage</code> .

<code>openleft</code>	<code>\cleardoublepageusingstyle</code> et <code>\cleardoublestandardpage</code> d'une part, <code>\cleardoubleplainpage</code> , <code>\cleardoubleemptypage</code> avec <code>\cleardoublepage</code> d'autre part, génèrent si besoin un saut de page et ajoutent un espace vide dans la présentation double face sur le côté gauche de pièce, chapitre, index et annexe <code>\cleardoublepage</code> .
<code>openright</code>	<code>\cleardoublepageusingstyle</code> et <code>\cleardoublestandardpage</code> d'une part, <code>\cleardoubleplainpage</code> , <code>\cleardoubleemptypage</code> avec <code>\cleardoublepage</code> d'autre part, génèrent si besoin un saut de page et ajoutent un espace vide dans le côté droit de la disposition double face de pièce, chapitre, index et espace <code>\cleardoublepage</code> .
<code>optiontohead</code>	La fonctionnalité avancée de l'argument des commandes de sectionnement à option est activée. L'argument optionnel est utilisé exclusivement pour le titre, par défaut.
<code>optiontoheadandtoc</code> , <code>optiontotocandhead</code>	La fonctionnalité avancée de l'argument des commandes de sectionnement est activée. Par défaut, l'argument optionnel est utilisé pour <code>Columnentitel</code> ainsi que l'entrée dans la table des matières.
<code>optiontotoc</code>	La fonctionnalité avancée de l'argument des commandes de sectionnement est activée. Par défaut, l'argument optionnel est utilisé exclusivement pour l'entrée dans la table des matières.
<code>small</code>	Utiliser des rubriques de petits intervalles au dessus et au dessous.
<code>twolineappendix</code> , <code>appendixwithprefix</code> , <code>appendixwithprefixline</code> , <code>appendixprefix</code>	Le titre des chapitres dans l'annexe est fixé avec une ligne d'entête dont le contenu est déterminé par <code>\chapterformat</code> .

-----  
`twolinechapter,`      Le titre des chapitres est fixé avec une ligne  
`chapterwithprefix,`    d'entête dont le contenu est déterminé par  
`chapterwithprefixline,` `\chapterformat.`  
`chapterprefix`

`numbers=cadre`

Selon DUDEN, seuls les chiffres arabes sont utilisés pour la numérotation sans point final pour clore le numéro de `section` (voir [DUD96R3]). En revanche l'utilisation de chiffres romains, de majuscules ou de minuscules nécessite l'obligation d'un point à la fin de tous les numéros d'unité (voir [DUD96R4]).

KOMA-Script a un mécanisme interne qui tente de répondre à ces règles quelque peu complexes. L'effet qui en résulte est que, normalement, après les commandes sectionnelles `\part` ou une pièce jointe (`\appendix`) à la numérotation, l'élément est commuté avec un point de conclusion. Ces informations sont stockées dans le fichier et prennent effet à la prochaine exécution de  $\LaTeX$  sur le document entier.

Parfois, avec une présélection de `numbers=autoendperiod` le programme automatique ne parvient pas à insérer ou retirer le point final dans le nombre de lignes et par conséquent, il est possible de prescrire, avec le réglage `numbers=endperiod`, le point manuellement ou de l'interdire avec `numbers=noendperiod`.

Il convient de noter que le mécanisme, lors de la prochaine exécution de  $\LaTeX$ , activera le point final ou le désactivera. En essayant de forcer la présentation correcte sur l'utilisation de l'une des options fondamentale, un autre terme  $\LaTeX$  doit être effectué sans changer.

Un résumé des valeurs possibles pour le réglage des numéros est dans la table 2.11 page 77. Contrairement à la plupart des autres paramètres, cette option ne peut être faite dans le préambule du document, c'est-à-dire avant `\begin document`.

`chapteratlists`  
`chapteratlists=valeur`

Comme il est mentionné à l'article 2.20, l'option `listof` ajoute à chaque entrée générée avec `\chapter` une distance verticale dans les listes de

TABLE 2.14 – Valeurs possibles de l’option pour sélectionner les intercalaires avec des pages blanches

---

<code>autoendperiod</code> , <code>autoenddot</code> , <code>auto</code>
KOMA-Script prend la décision du numéro de l’article et du nombre de numéros à charger Il utilisera pour tous les articles des chiffres arabes. Si un numéro d’unité est une lettre ou un chiffre romain, ce point sera réglé pour tous les numéros et les références à ces chiffres seront fixées dès le départ.
<code>endperiod</code> , <code>withendperiod</code> , <code>periodatend</code> , <code>enddot</code> , <code>withenddot</code> , <code>dotatend</code>
Pour les numéros d’unité et à charge omis lors du référencement.
<code>noendperiod</code> , <code>noperiodatend</code> , <code>noenddot</code> , <code>nodotatend</code>
Les numéros hiérarchiques et les numéros dépendants sont fixés sans point de fuite.

---

flottants qui s’applique également, depuis la version 2.96a à l’expression `\addchap` si aucun paramètre de compatibilité pour une version antérieure n’a été sélectionné (voir la version de l’option dans la [section 2.2](#)).

La distance peut également être modifiée avec l’option `chapteratlists` Pour cela, on l’ajoute en tant que valeur de la distance souhaitée. Avec le réglage par défaut `listof=chaptergapsmall` la distance est de 10 pt.

Avec le réglage `chapteratlists=entry` ou en utilisant le formulaire `chapteratlists` sans entrée, la valeur est indiquée dans les répertoires au lieu de la distance. Cette entrée a lieu également lorsque le chapitre ne contient pas d’environnement flottant. Noter que pour un changement de ce type, la modification nécessite une double compilation.

```
\part[forme abrégée]{titre}
\chapter[forme abrégée]{titre}
\section[forme abrégée]{titre}
\subsection[forme abrégée]{titre}
\subsubsection[forme abrégée]{titre}
\paragraph[forme abrégée]{titre}
\subparagraph[forme abrégée]{titre}
```

La section des commandes dans KOMA-Script est, en principe, comme dans la classe `standard`. Ainsi, normalement via un argument optionnel un texte de remplacement peut être spécifié, par défaut, pour le titre de la colonne et la table des matières.

Avec les rubriques de réglage `headings=optiontohead`, un argument optionnel KOMA-Script ne peut être utilisé que pour le titre en cours d'exécution. Une entrée spécifique dans le titre de la colonne est utilisée pour la manchette si un style est choisi avec un niveau de détail approprié. Voir détails section 2.12, chapitre 5 et chapitre 6.

Avec le paramètre `headings=optiontotoc` l'argument optionnel agit en exclusivité pour l'entrée dans la table des matières. Cela, aussi, bien sûr, lorsque le réglage du compteur `tocdepth` (voir section 2.9) fournit une liste de niveau approprié à tous.

Avec le réglage `headings=optiontoheadandtoc` l'argument optionnel est utilisé par le titre de la colonne et l'entrée de la table des matières. Les trois paramètres permettent l'interprétation extensive de l'argument par opposition à la valeur par défaut. L'interprétation de l'argument cochée indique un signe égal de forme courte et dans ce cas, l'argument pour les commandes de sectionnement est interprété comme une forme courte et placé dans une liste d'options. Les options `head=Kolummentitel` et `tocentry=entrydetabledesmatières` seront acceptées.

Mettre un signe égal ou une virgule dans l'une des valeurs de ces options doit se faire entre accolades supplémentaires. Ce mécanisme ne fonctionne que tant que KOMA-Script exerce un contrôle sur les sectionnements. Si un paq utilise ces commandes ou un état du noyau L<sup>A</sup>T<sub>E</sub>X interne pour ces mêmes commandes redéfinies, KOMA-Script peut ne plus fournir cette fonctionnalité élargie.



Cela est également vrai pour l'extension d'entrée de contenus avec table vide qui conduit à une entrée sans texte complètement omise. En effet, une entrée vide affecte la table des matières, mais peut être incluse avec une entrée invisible comme `\mbox`.

Supposons que vous avez un document avec quelques entêtes de chapitre très longs. Ils seront inclus dans la table des matières mais vous pouvez en limiter la longueur du texte. Il suffit de mettre :

```
\documentclass[headings=optiontohead]{scrbook}
```

Lors du chargement de la classe, l'argument optionnel des commandes de sectionnement sera utilisé pour le titre et, dans le document une entrée correspondante créée avec l'argument optionnel de `\chapter` :

```
\chapter[format abrégé pour chapitre]
{La structure de commandement pour le titre des chapitres
permet de gérer un texte pour une forme courte avec utilisati-
on contrôlable}.
```

Un peu plus tard vous constatez que ce long titre ne doit pas être rompu et vous souhaitez en déterminer le graphisme dans l'entête lui-même, mais la table des matières continue d'être créée automatiquement. Avec :

```
\chapter[head={format abrégé pour chapitre},
tocentry = {la structure de commande pour le titre des
chapitres permet non seulement un ajout bienvenu au
texte pour le réel, le titre de chapitre, mais aussi
une forme courte avec une utilisation contrôlable.
{La structure de commande pour le titre des chapitres \\
permet non seulement un ajout bienvenu au texte \\
pour le réel, le titre de chapitre, \\
mais aussi une forme courte \\
avec une utilisation contrôlable}
```

Les entrées sont redéfinies pour le titre de colonne et la table des matières indépendamment de la position elle-même puis vous ajoutez les arguments des deux options `head` et `tocentry` entre accolades de sorte que le contenu des arguments peut être aléatoire.

Parfois, vous prenez conscience que la rupture automatique de ligne de cette rubrique est inappropriée, et vous voulez la faire vous-même. Mais

néanmoins, la rupture de la ligne automatique sera toujours utilisée dans la table des matières avec `KOMA-Script`.

La nécessité des accolades peut être mieux illustrée par un exemple. Supposons que `headings=optiontotoc` soit choisie comme option dont il faut définir l'écriture :

```
\section[head=\emph{valeur}]{l'option head=emph{valeur}}
```

Cela signifie que le contenu de l'entrée `option head=valeur` et celui du titre de la colonne de l'entrée `valeur` seront utilisés. En réalité, vous vouliez que le contenu de l'entrée `head=valeur` soit adopté dans le titre de la colonne du texte du titre. Ceci peut être réalisé en entourant le signe égal d'accolades :

```
\section[head={}\emph{valeur}]{l'option head=\emph{valeur}}
```

Un cas similaire concerne la virgule. Parce que les mêmes options par défaut :

```
\section[head=0, 1, 2, 3,\dots]{les nombres naturels avec zéro}
```

conduisent à un message d'erreur, car les virgules seraient interprétées comme séparateurs entre les options de la liste des options `head=0, 1, 2, 3,\dots` serait interprété ainsi. Mais si vous écrivez :

```
\section[head={0, 1, 2, 3, \dots}]{les nombres naturels avec zéro}
```

alors `0, 1, 2, 3,\dots` deviennent un argument de l'option `head`.

La position du niveau de (`\part`) diffère des autres niveaux structurés en ce qu'il est numéroté indépendamment des autres plan, cela signifie que chaque niveau d'un chapitre (`scrbook` ou `scrreprt`) ou d'une section (`scrartcl`) est numéroté en continu sur toutes les parties extérieures.

En outre, il existe pour les classes `scrbook` et `scrreprt` le titre de ce plan ainsi que son préambule (voir la déclaration `\Setpartpreamble`) `\chapter` existe seulement dans les librairies et les classes rapports qui comprennent `book`, `scrbook`, `report` et `scrreprt`, mais pas dans l'article des classes `article` et `scrartcl`. `\chapter` dans `KOMA-Script` diffère également de la version des classes `standard`. Dans ces classes, le numéro du

chapitre est publié avec le préfixe `chapter` et le nom du chapitre dans la langue sélectionnée une ligne avant le texte de la rubrique. Cette forme très puissante sera remplacée avec `KOMA-Script`, par un simple numéro avant le texte mais les options peuvent être redéfinies avec `\chapterprefix` et `\appendixprefix`. Il faut noter que dans `scrbook` et `scrreprt`, `\part` et `\chapter` permettent de commuter le style de page.

Un style particulier de page est défini dans les macros `\partpagestyle` et `\chapterpagestyle` (voir la section 2.12). La fonte des sept niveaux hiérarchiques peut être déterminée avec les commandes `\setkomafont` et `\addtokomafont` (voir la section 2.6), premier paramétrage utilisé dans la disposition générale par niveau et la mise en place de l'élément spécifique, `disposition` (voir la table 2.2 page 45). Sa police est prédéfinie comme `\normalcolor`, `\sffamily` ou `\bfseries`. Les paramètres par défaut pour les éléments spécifiques sont initialisés comme options de réglage de la taille de police `\big`, `\normal` ou `\small`. (paramètres par défaut dans la table 2.15 page 95).

Vous choisissez d'utiliser les rubriques d'options de classe `headings=big` avec de très grands titres en gras? Utiliser :

```
\setkomafont{disposition}{\normalcolor\sffamily}
\part{appendices}
\addtokomafont{disposition}{\bfseries}
```

Pour modifier cela, procédez comme suit et de cette façon, vous supprimez l'attribut de police gras pour une rubrique **Annexe**. Une solution beaucoup plus confortable et élégante serait d'utiliser `\part` avec un changement correspondant. Cela est possible et se ferait par :

```
\addtokomafont{part}{\normalfont\sffamily}
\addtokomafont{partnumber}{\normalfont\sffamily}
```

ou encore plus simplement avec :

```
\addtokomafont{part}{\mdseries}
\addtokomafont{partnumber}{\mdseries}
%}
```

L'utilisation de `\setkomafont` serait en principe possible, il contiendrait l'option du choix de la taille mais aussi l'option du redimensionnement.

La version avec `\mdseries` est préférable parce que cela conduira au résultat souhaité, si vous changez l'élément `disposition` comme suit :

```
\setkomafont{disposition}{\normalcolor\bfseries}
```

Dans cette version, il est préférable de s'abstenir d'utiliser une police sans serif préreglée pour tous les niveaux hiérarchiques. Ainsi, je vous conseille de conserver la possibilité de changer la police et en cas de nécessité de mélanger les polices, les tailles et les attributs ensemble. Choisir la bonne police pour le bon emploi est une question d'expert et n'a rien à voir avec le goût personnel d'un débutant. Voir la citation à la fin de la **section 2.8** et la renonciation suivante.

Différentes polices pour différents niveaux de détails sont possibles mais le néophyte, pour des raisons de typographie, doit les éviter comme la peste.

Une règle stipule que vous devez mélanger aussi peu que possible les polices ensemble. **sans serif** pour les titres semble déjà être une violation de cette règle. Toutefois, vous devez savoir que les lettres audacieuses, énormes avec empattement sont beaucoup trop puissantes pour un titre. On aurait alors intérêt à passer une normale à la place d'une police en gras ou semi-gras.

Ce sont des détails et pourtant, la police **sans serif** donne un aspect très agréable à un titre et ne devrait être utilisée que pour les rubriques autorisées. Par conséquent, cette valeur par défaut pour **KOMA-Script** a été choisi avec raison.

Plus de variété doit être évitée. Les mélanges de polices restent seulement pour les professionnels. Pour ces raisons, vous pouvez utiliser autre chose que la norme **TEX-Fonts**, peu importe si les polices **cm**, **ce** ou **lm**, en ce qui concerne la compatibilité avec **sans serif** et **serif**, demandent l'avis d'un expert. Les combinaisons fréquemment rencontrées avec **Helvetica** ou **Palatino** sont considérées par l'auteur comme défavorables.

TABLE 2.15 – Paramètres par défaut de la police pour les éléments de la structure de `scrbook` et `scrreprt`

Option de classe	Élément	Par défaut
<code>headings=big</code>	<code>part</code>	<code>\Huge</code>
	<code>partnumber</code>	<code>\huge</code>
	<code>chapter</code>	<code>\huge</code>
	<code>section</code>	<code>\Large</code>
	<code>subsection</code>	<code>\large</code>
	<code>subsubsection</code>	<code>\normalsize</code>
	<code>paragraph</code>	<code>\normalsize</code>
	<code>subparagraph</code>	<code>\normalsize</code>
<code>headings=normal</code>	<code>part</code>	<code>\huge</code>
	<code>partnumber</code>	<code>\huge</code>
	<code>chapter</code>	<code>\LARGE</code>
	<code>section</code>	<code>\Large</code>
	<code>subsection</code>	<code>\large</code>
	<code>subsubsection</code>	<code>\normalsize</code>
	<code>paragraph</code>	<code>\normalsize</code>
	<code>subparagraph</code>	<code>\normalsize</code>
<code>headings=small</code>	<code>part</code>	<code>\LARGE</code>
	<code>partnumber</code>	<code>\LARGE</code>
	<code>chapter</code>	<code>\Large</code>
	<code>section</code>	<code>\large</code>
	<code>subsection</code>	<code>\normalsize</code>
	<code>subsubsection</code>	<code>\normalsize</code>
	<code>paragraph</code>	<code>\normalsize</code>
	<code>subparagraph</code>	<code>\normalsize</code>

```
\part*{titre}
\chapter*{titre}
\section*{titre}
\subsection*{titre}
\subsubsection*{titre}
\paragraph*{titre}
\subparagraph*{titre}
```

La version d'une commande étoilée ne comporte aucune numérotation et crée un titre de colonne qui ne comporte pas d'entrée dans la table des matières et se fier à ce titre a souvent un effet indésirable. Par exemple, si un `\chapter*` comporte des chapitres sur plusieurs pages, c'est souvent le titre usuel du dernier chapitre qui sera mis en avant. **KOMA-Script** offre une solution décrite par la suite. `\chapter*` existe dans `book`, `scrbook`, `report` et `scrreprt`, mais pas dans les classes `article` et `scrartcl`.

Noter que `\part*` et `\chapter*` changent, par commutation, le style de page. Bien qu'utilisant les classes `standard`, le style `plain` est le style de page à utiliser dans les macros `\partpagestyle` et `\chapterpagestyle` avec **KOMA-Script** (voir l'article 2.12).

**KOMA-Script** applique, pour les possibilités de commutation de la police, le même principe que celui précédemment décrit pour les déclarations de variantes sans étoiles. Les éléments ont les mêmes noms car ce ne sont pas des variantes mais ils décrivent les niveaux.

Dans les classes `standard`, des commandes de création automatique du titre existent. Il n'y a donc pas de consignes particulières quant au chiffre du chapitre numéroté ou non inclus dans la table des matières.

```
\addpart{raccourci}{titre}
\addpart*{titre}
\addchap{raccourci}{titre}
\addchap*{titre}
\addsec{raccourci}{titre}
\addsec*{titre}
```

**KOMA-Script** intègre les commandes de classes `\addpart`, `\addchap` et `\addsec`, au fonctionnement similaires des commandes standard `\part`, `\chapter` et `\section` : absence de numérotation à l'entrée dans la table des matières, en respectant les paramètres de l'option `titres`.

Les versions étoilées `\addchap*` et `\addsec*` équilibrent les commandes `\chapter*` et `\section*` avec une différence minuscule mais importante : le titre courant sera supprimé.

Ainsi, l'effet mentionné exclut les entêtes obsolètes. Au lieu de cela, les rubriques de pages suivantes seront vides. `\addchap` et `\addchap*` bien sûr n'existent que dans les bibliothèques et les classes rapports qui comprennent `scrbook` et `scrreprt`, mais pas dans `scrartcl`.

La commande `\addpart` crée un document non numéroté avec une entrée dans la table des matières. Comme `\part` et `\part*` supprime le titre de la colonne, le résultat ici n'est pas le problème des titres de colonnes obsolètes. La version étoilée `\addpart*` est identique à la version étoilée `\part*` et n'est définie que pour des raisons de cohérence.

Noter que `\addpart`, `\addchap` et leurs versions étoilées changent le style de page pour la page en cours. Le style particulier de page est défini (voir la section 2.12) dans les macros `\partpagestyle` et `\chapterpagestyle`.

Les possibilités de commutation sont identiques dans les déclarations écrites pour `\part`, `\chapter` et `\section*`. Les éléments ont les mêmes noms car ils ne font pas des variantes mais décrivent les niveaux.

`\minisec{titre}`

Parfois, une sorte de titre est souhaitable, mis en évidence ou fortement lié au texte suivant. Une telle position ne doit pas être séparée par une verticale.

La commande `\minisec` est conçue pour cela. Elle n'est pas destinée à décrire les niveaux et n'est pas incluse dans la table des matières, donc ne reçoit aucune numérotation.

La police de la commande de structuration `\minisec` peut être gênée par les éléments `disposition` et `minisec v2.98ab` (voir la table 2.2 page 45). Par défaut, `minisec` est vide et n'a pas de valeur.

Exemple : vous avez développé un kit pour un piège à souris et vous voulez en décrire les matériaux nécessaires ainsi que les instructions de montage. Cela pourrait être fait avec :

```
\documentclass{scrartcl}
\usepackage[french]{babel}
```

```
\usepackage{selinput}
\SelectInputMappings{
    adieresis={ä},
    germandbls={ß}
}
\begin{document}
\title{Projet de bricolage}
\author{Deux pouces gauches}
\date{\today}
\maketitle
\section{Piège à souris}
Le premier projet est adapté pour les débutants et ne nécessite
que quelques composants qui devraient être trouvés dans tous
les foyers.
\minisec{Composants}
\begin{aligné à gauche}
1 planche (100\times 50 \times 12$)
1 bouteille de bière avec fermeture à pression
stylo bille
punaise
2 vis
marteau
couteau
\end{aligné à gauche}
\minisec{Montage}
Pour commencer, cherchez le trou de souris et mettez la pu-
naise derrière, afin que la souris ne puisse pas s'échapper
en attendant la suite des opérations.
Enfoncez à l'aide d'un marteau la bouteille de bière dans le
trou. Si l'obturateur n'est pas assez grand pour le fermer,
prenez la planche, fixez la avec les deux vis et le couteau.
Au lieu du couteau vous pouvez utiliser un tournevis.
Le stylo permet de noter le nombre de victimes capturées.
\end{document}
```

Cependant, la majeure partie de la rubrique « composants » ressemble à ce qui suit :



1 planche (100 × 50 × 12) 1 bouteille de bière avec fermeture à pression stylo bille punaise - 2 vis marteau - couteau
--

Pour comprendre l'utilisation de `selinput` et de `\SelectInputMappings`, nous nous référons à [Obe07].

<code>\raggedsection</code> <code>\raggedchapter</code> <code>\raggedpart</code>
--

Dans les classes `standard` les rubriques sont émises en un seul bloc. En conséquence, les séparations peuvent se produire dans les titres et entêtes qui s'étendent sur plusieurs lignes à la largeur du texte. C'est assez rare en typographie et KOMA-Script utilise de `\chapter` à `\ subparagraph` un alinéa aligné à gauche par défaut, avec un retrait négatif dû à la commande prédéfinie `\raggedsection` :

```
\let\raggedsection\raggedright
```

La commande `\raggedsection` est redéfinie avec `\renewcommand`.

Pour avoir des rubriques justifiées, vous écrivez dans le préambule de votre document :

```
\newcommand*{\raggedsection}
```

ou plus court :

```
\let\raggedsection\relax
```

Ainsi vous obtiendrez un formatage des entêtes comme dans les classes `standard` et même plus en combinant ce changement avec le changement pré-établi pour l'élément. Les titres des parties (`\part`) sont définis par défaut comme les seuls non alignés à gauche mais centrés par la commande `\raggedpart` :

```
\let\raggedpart\centering
```

prédéfinie, mais qui peut être redéfinie avec `\renewcommand`. Pour formater les titres avec `\part` de la même manière que tous les autres niveaux, vous écrivez dans le préambule de votre document :

```
\renewcommand*{\raggedpart}{\raggedsection}
```

À ce stade `\let` n'est pas utilisé pour ne pas modifier `\raggedpart`, ce qui affecterait `\raggedsection` dont les variations ultérieures resteraient méconnues de `\raggedpart`. Après une redéfinition avec `\renewcommand`, `\raggedpart` ne reçoit pas le sens de `\raggedsection` au moment de cette redéfinition, mais celui de `\raggedpart` au moment de son utilisation.

KOMA-Script ajoute la sortie des numéros de `section` au dessus du nom de structure (voir le compteur) pour sauvegarder un niveau plus logique. Les compteurs ne sont pas seulement réservés aux titres.

Ils utilisent les commandes `\partformat`, de `\chapterformat` jusque `\subparagraphformat` sans paramètres et attendent trois paramètres avec `\othersectionlevelsformat` de `scrbook`.

```
\partformat
\chapterformat
\sectionformat
\subsectionformat
\subsubsectionformat
\paragraphformat
\subparagraphformat
\othersectionlevelsformat{structure}{sortie compteur}
\IfUsePrefixLine{alors}{sinon}
\autodot
```

La déclaration `\chapterformat` existe dans `\thechapter` les classes `scrbook` et `scrreprt` mais pas dans la classe `scrartcl`.

Comme avec l'option de numéros expliquée au début de cette `section`, selon [DUD96] un numéro hiérarchique est estampillé selon la `section` avec un point de fuite, ou doit être éliminé. La commande `\autodot` dans KOMA-Script est responsable du respect de cette règle. À l'exception de `\part`, au point succède toujours un `\enskip` à tous les niveaux de détail, ce qui correspond à une espace de 0,5 em.

La déclaration `\othersectionlevelsformat` prévoit le nom du niveau hiérarchique comme premier paramètre, c-à-d la `section`, `sous-section`, `sous-sous-section`, le paragraphe ou l'alinéa.

Le troisième argument est la sortie prévue du compteur au niveau de `\thesection`, `\thesubsection`, `\thesubsubsection`, `\theparagraph` ou `\thesubparagraph`. Par défaut, les niveaux `\part` et `\chapter` comportent seuls leurs propres instructions de mise en forme, alors que tous les autres niveaux hiérarchiques sont couverts par un format unique.

Cela est justifié historiquement. Werner LEMBERG a développé, pour son paq CJK, (voir [Lem08]) une expansion correspondante de KOMA-Script, différenciation jugée nécessaire. Les utilisateurs doivent ignorer le deuxième argument complètement dans la redéfinition de la déclaration.

À l'aide de `\renewcommand`, chacune des trois commandes de formatage peut être redéfinie pour correspondre à vos besoins. Vous trouverez les définitions originales des classes KOMA-Script ci-dessous :

```
\newcommand*{\partformat}{\partname~\thepart\autodot}
\newcommand*{\chapterformat}{%
\mbox{\chapappifchapterprefix{\nobreakspace}\the chapter
\autodot \IfUsePrefixLine{}{\enskip}}}
\newcommand*{\sectionformat}{\thesection\autodot\enskip}
\newcommand*{\othersectionlevelsformat}[3]{%
#3\autodot\enskip}
```

Les niveaux plus profonds sont définis en conformité avec `\sectionformat`.

Avec l'utilisation de `\IfUsePrefixLine` `\chapterformat` ne devrait pas être utilisé en dehors. Cette déclaration est seulement définie dans les grandes lignes de commandes KOMA-Script. Dans le cas de l'utilisation d'un préfixe pour le numéro de ligne de l'entête, la condition « alors » est exécutée, et dans tous les autres cas, la commande « sinon » est activée par défaut.

N'oubliez pas de remplacer, pour une redéfinition, `\newcommand` par `\renewcommand`.

Si vous voulez que dans `\part` le mot partie ne sorte pas avant le numéro, vous pouvez écrire, par exemple, la déclaration suivante dans le préambule de votre document :

```
\renewcommand*{\partformat}{\thepart\autodot}
```

En fait, vous pouvez, ici, faire sans `\autodot` et définir un point fixe.

Comme la pièce `\part` est numérotée avec des chiffres romains, le point selon [DUD96] doit suivre. Cependant, vous devrez avoir la possibilité d'utiliser les numéros d'option et ainsi de déroger à la règle.

Une autre option consiste à placer, alignés à gauche comme le texte d'entête, les numéros de `section` dans la marge, entourés par le texte. Ce qui est obtenu avec :

```
\renewcommand*{\sectionformat}{%
\makebox[0pt][r]{\thesection\autodot\enskip}}
\renewcommand*{\subsectionformat}{%
\makebox[0pt][r]{\thesubsection\autodot\enskip}}
\renewcommand*{\subsubsectionformat}{%
\makebox[0pt][r]{\thesubsubsection\autodot\enskip}}
\renewcommand*{\paragraphformat}{%
\makebox[0pt][r]{\theparagraph\autodot\enskip}}
\renewcommand*{\paraglyphformat}{%
\makebox[0pt][r]{\thesubparagraph\autodot\enskip}}
```

Vous trouverez les détails sur les arguments optionnels de `\makebox` en consultant [Tea05b].

Pour le formatage des modifications de comparaison des entêtes qui vont au-delà de l'addition de nombre, voir `\chapterlineswithprefixformat`, `\chapterlinesformat`, `\sectionlinesformat` rapportés à la section 21.3 ainsi que `\sectioncatchphraseformat`.

```
\chapappifchapterprefix{texte complémentaire}
\chapapp
```

Ces deux commandes sont utilisées en interne par KOMA-Script, mais également disponibles pour l'utilisateur. Par exemple, les commandes ci-dessous permettront de définir d'autres instructions.

Si vous utilisez `\chapappifchapterprefix` avec `chapterprefix=true` comme option dans la partie principale du document, le mot `Chapitre` est suivi par du texte supplémentaire. Dans l'annexe, à la place du mot `Annexe` exprimé dans le langage courant est émis un texte supplémentaire. L'option `chapterprefix=false` bloque cette émission.

La commande `\chapapp` émet, par défaut, les mots `Chapitre` ou `Annexe`. Le réglage de l'option `chapterprefix` n'a donc aucune importance.

Ces deux commandes, comme les chapitres, n'existent que pour les classes `scrbook` et `scrreprt`.

```
\chaptermark{titre courant}
\addchapmark{titre courant}
\sectionmark{titre courant}
\addsecmark{titre courant}
\subsectionmark{titre courant}
\chaptermarkformat
\sectionmarkformat
\subsectionmarkformat
```

Comme il est mentionné à la section 2.12, les entêtes de page `headings` fonctionnent avec un style automatique de titre de colonne et pour ce faire, les commandes `\chaptermark` et `\sectionmark` définissent, en ce qui les concerne, `\sectionmark` et `\subsectionmark` en conséquences. Les commandes de sectionnement `\chapter`, `\section...` effectuent leur action automatiquement dans la commande `\...-mark` correspondante. Le texte de l'entête de la rubrique est inclus dans le paramètre passé. Le numéro de l'article correspondant est automatiquement ajouté à la commande `\...mark`. La mise en forme se fait selon le niveau de la section avec l'une des commandes `\textchaptermarkformat`, `\textsectionmarkformat` et `\textsubsectionmarkformat`.

Depuis KOMA-Script les instructions `\addchap` et `\addsec` produisent des rubriques non numérotées qui correspondent aux instructions pour le titre. Noter que ces accès par défaut `\chaptermark` et `\sectionmark`, mettent en place un compteur `secnumdepth` à -1 avec lequel le foliotage des chapitres et sections est désactivé, ce qui est pris en compte dans la redéfinition de `\chaptermark` et `\sectionmark` (voir `\ifnumbered`).

Pour les formes étoilées, `\addchap*` et `\addsec*` il existe les commandes `\addchapmark` et `\addsecmark` qui sont également définies par défaut de la manière ci-dessus.

Dans `scrartcl`, la commande `\chaptermark` ou `\addchapmark` n'existe pas, ni `\chaptermarkformat` mais deux commandes `\subsectionmark` et `\subsectionmarkformat` existent dans `scrartcl`. Les choses changent lorsque vous utilisez les paqs `\sclayer-scrpage` ou `\scrpage2` (voir chap. 5 et 6).

Tout comme avec `\partformat` et `\subparagraphformat` les numéros des rubriques hiérarchiques sortent formatés.

Le détail du nombre de niveaux de sorties dans les rubriques de mise en forme automatique du titre des colonnes est indiqué par les commandes `\chaptermarkformat`, `\sectionmarkformat` et `\subsectionmarkformat`. Elles peuvent, avec `\renewcommand`, être personnalisées à vos besoins. Les définitions originales des classes KOMA-Script sont :

```
\newcommand*{\chaptermarkformat}{%
\chapappifchapterprefix{\ } \thechapter \autodot \enskip}
\newcommand*{\sectionmarkformat}{%
\thesection \autodot \enskip}
\newcommand*{\subsectionmarkformat}{%
\thesubsection \autodot \enskip}
```

Pour que le numéro de chapitre soit précédé par le mot chapitre dans les titres des colonnes, régler cette définition dans le préambule de votre document :

```
\renewcommand*{\chaptermarkformat}{%
\chapapp~ \thechapter \autodot \enskip}
```

Comme vous pouvez le voir l'utilisation des deux commandes `\chapapp` et `\chapappifchapterprefix` est décrite ci-dessus.

<pre>secnumdepth \partnumdepth \chapternumdepth \sectionnumdepth \subsectionnumdepth \subsubsectionnumdepth \paragraphnumdepth \subparagraphnumdepth</pre>
--

Les niveaux hiérarchiques des classes `scrbook`, `scrreprt`, `\part` jusque `\subsection` et de la classe `\scrartc`, de `\part` jusque `\subsubsection` ont des paragraphes normalement numérotés, numérotation contrôlée par la commande `\secnumdepth` du compteur de L<sup>A</sup>T<sub>E</sub>X.

Cela signifie que l'utilisateur ne doit pas noter de nombres abstraits. Pour sélectionner à quel niveau les rubriques doivent être numérotées,

les commandes `\partnumdepth` et `\subparagraphnumdepth` fournissent le nombre approprié pour chaque niveau hiérarchique, système applicable également dans les classes `standard`. Vous trouverez plus de détails sur le compteur `tocdepth` dans la section 2.9.

```
\ifnumbered{niveau}{alors}{sinon}
\ifunnumbered{niveau}{alors}{sinon}
```

Ayant déjà expliqué comment déterminer les niveaux hiérarchiquement numérotés, ces instructions fonctionnent si un niveau de plan est numéroté ou pas. Avec les paramètres actuels, `\ifnumbered` agit sur la partie alors que `\ifunnumbered` exécute la partie else. Non numéroté, c'est l'inverse qui se produit.

Le nom anglais de niveau n'est pas spécifié, `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` ou `subparagraph`. KOMA-Script lui-même utilise cet essai, par exemple dans la définition des rubriques de style de page `\chaptermarkformat` et indirectement assure également les titres numérotés de colonne avec `\addchap` (voir aussi `\addchapmark`).

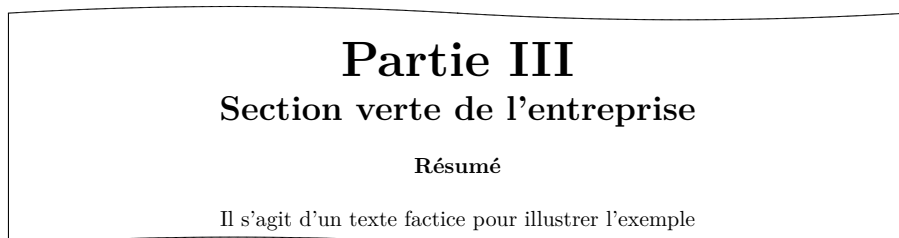
```
\setpartpreamble{Position}{largeur}{préambule}
\setchapterpreamble{Position}{largeur}{préambule}
```

Chapitres et parties peuvent être fournis avec un préambule, la mise en page deux colonnes avec l'option de classe `twocolumn` en particulier. Le préambule occupe une colonne avec l'entête et peut également inclure plusieurs graphiques. La commande de l'émission du préambule précède respectivement chaque `\part`, `\addpart`, `\chapter` ou `\addchap`.

Si pour rédiger un rapport sur l'état d'une société vous l'organisez de manière à ce que chaque département intègre son propre rapport partiel, précédé d'un résumé (`abstract`) qui apparaîtra sur chaque page de titre, ce qui est rendu possible de la manière suivante :

```
\setpartpreamble{%
\begin{abstract}
    Il s'agit d'un texte factice pour illustrer l'exemple
\end{abstract}
\part{Section verte de l'entreprise}
```

Selon les paramètres de la taille de l'entête et les options pour la forme de l'environnement `abstract` (voir [section 2.8](#)), le résultat ressemblera à l'image suivante.



Noter que vous êtes responsable de la distance entre le préambule de la sous-rubrique et le texte suivant et qu'il n'existe pas d'environnement `abstract` dans la classe `scrbook` (voir [section 2.8](#)).

La position du premier argument optionnel détermine suivant deux lettres, la position à laquelle le préambule est émis. Pour le placement vertical :

- **o** sur le titre,
- **u** sous la rubrique,  
et pour la position horizontale :
- **l** aligné à gauche,
- **r** aligné à droite,
- **c** centré.

Dans ce cas cependant, le texte du préambule utilise une boîte dont la largeur est déterminée par la largeur du second argument optionnel. Si on omet cet argument, toute la largeur du texte est utilisée et l'option pour le positionnement horizontal devient inefficace. Vous pouvez insérer une lettre pour la position verticale combinée avec une lettre de la position horizontale. Une utilisation plus fréquente de `\setchapterpreamble` est susceptible de gérer le titre d'un dicton ou d'une citation. On utilisera alors la commande `\dictum`.

Il est à noter que la mise du préambule sur le titre occupe l'espace libre existant. Le titre n'est pas décalé vers le bas. L'utilisateur est donc tenu de s'assurer que le préambule n'est pas trop grand et satisfait l'espace libre pré-existant au-dessus de l'entête, voir aussi `\chapterheadstartvskip` au paragraphe 20.3.



## 2.17 Citation et dicton

Un élément fréquemment rencontré est une citation ou une phrase qui est placée juste en dessous ou au-dessus d'un titre. Ici, la *sentence* elle-même et la mention de la source sont généralement spécialement formatées.

Cette « maxime » peut être réglée en utilisant `\dictum` comme argument obligatoire des commandes `\setchapterpreamble` ou `\setpartpreamble` pour chapitre et une partie des classes KOMA-Script, (voir section 2.16). La largeur du dicton est réglée à l'aide de `\dictumwidth`. Le dicton avec un auteur en option est inséré dans un `\parbox` (voir [Tea05b]). Mais `\dictumwidth` n'est pas une longueur qui peut être réglée avec `\setlength`. Il s'agit d'une macro qui peut être redéfinie en utilisant `\renewcommand`. Le réglage `\textwidth` par défaut est 0.3333, soit un tiers de la largeur du texte.

La boîte est positionnée avec la commande `\raggeddictum`. Par défaut `\raggedleft` est justifié à droite. La commande `\raggeddictum` peut être redéfinie à l'aide de `\renewcommand`.

Cet effet peut être réglé en utilisant la commande `\dictum`. Dans les classes KOMA-Script, il est recommandé pour les chapitres ou parties.

```
\dictum{créateur}{maxime}
\dictumwidth
\dictumauthorformat{créateur}
\dictumrule
\raggeddictum
\raggeddictumtext
\raggeddictumauthor
```

TABLE 2.16 – Paramètres par défaut de la police pour les éléments de la maxime

Élément	Paramètres par défaut
<code>dictumtext</code>	<code>\normalfont\normalcolor\sffamily\small</code>
<code>dictumauthor</code>	<code>\itshape\medbreak</code>

La maxime dans la boîte est définie en utilisant `\raggeddictumtext`. Le réglage par défaut est `\raggedright`, justifié à gauche. De même pour `\raggeddictum` qui peut être redéfini avec `\renewcommand`.

La sortie utilise pour `dictumtext` les paramètres de l'élément police par défaut. Cette dernière peut être modifiée avec la commande `\setkomafont` et la commande `\addtokomafont` (voir section 2.6). Les réglages par défaut sont répertoriés dans la table 2.16 page précédente. Si un nom d'auteur est indiqué, il est séparé de la maxime par une règle sur toute la largeur de la `parbox`. Cette règle est définie comme objet vertical avec la commande `\dictumrule` :

```
\newcommand*{\dictumrule}{\vskip-1ex\hrulefill\par}
```

L'ajustement de la ligne avec `\raggeddictumauthor` est pris avant l'auteur. La valeur par défaut est `\raggedleft`. Cette commande peut être redéfinie avec `\renewcommand`. La sortie est dans la forme prévue avec `\dictumauthorformat`. La macro qui attend l'auteur comme argument par défaut est :

```
\newcommand*{\dictumauthorformat}[1]{(#1)}
```

L'auteur est entre parenthèses. Une police différente peut être définie pour l'élément `dictumauthor` avec `dictumtext`, voir à ce propos le réglage par défaut dans la table 2.16 page précédente. Un changement est possible avec les commandes `\setkomafont` et `\addtokomafont` (voir la section 2.6). Si une maxime est utilisée dans la déclaration `\setchapterpreamble` ou `\setpartpreamble`, le positionnement horizontal est toujours fait par `\raggeddictum` et les commandes de cet argument optionnel n'ont pas d'effet. La largeur de l'ensemble du corps de texte n'est pas `\textwidth` mais l'actuelle largeur de texte comme avec `minipage`.

Ainsi, si la largeur `\textwidth` de `\dictumwidth` est définie à 0.5 et que l'argument optionnel `\textwidth` de `\setchapterpreamble` spécifie aussi une largeur de 0.5, la sortie est dans une boîte dont la largeur est d'un quart de la largeur du corps de texte. Il est recommandé, si vous utilisez l'option `\dictum` de renoncer à l'option largeur de `\setchapterpreamble` ou de `\setpartpreamble`.

Plusieurs mots placés les uns sous les autres seront séparés verticalement par une distance supplémentaire qui peut être facilement réglée avec la commande `\bigskip`.

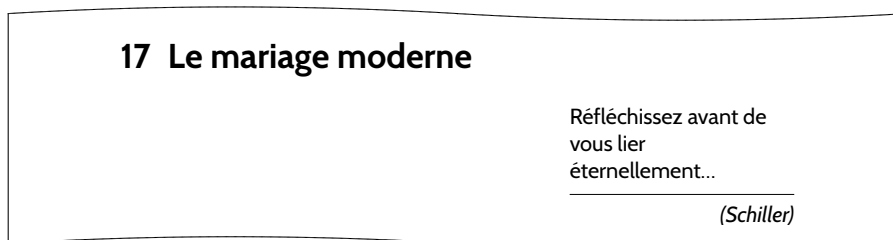
Pour rédiger un chapitre sur le mariage moderne et mettre un dicton intelligent dans le préambule qui apparaît sous le titre du chapitre, vous pourriez écrire :

```
\setchapterpreamble[u]{%
\dictum[Schiller]{Réfléchissez avant,
de vous lier éternellement \dots}}
\chapter{Le mariage moderne}
```

Si vous ne voulez pas du tiers, mais seulement du quart de la largeur du texte utilisé pour le dicton, redéfinir `\dictumwidth` comme suit :

```
\renewcommand*{\dictumwidth}{.25 \textwidth}
```

La sortie se fera sous la forme :



Pour une mise en forme un peu plus sophistiquée, paragraphes alignés à gauche ou à droite, avec ou sans césure, vous pouvez utiliser le paquet `ragged2e` [Sch09].

## 2.18 Listes

$\text{\LaTeX}$  et les classes `standard` offrent différents environnements pour les listes, tous `KOMA-Script`, bien sûr, mais parfois avec des modifications ou améliorations. Fondamentalement, les listes, même de nature différente, peuvent être imbriquées jusqu'à une profondeur de quatre. Une mosaïque plus profonde n'est guère souhaitable pour des raisons typographiques et

trois niveaux semblent largement suffisants. Il est donc recommandé, dans ce cas, de diviser une grande liste en listes plus petites.

La forme la plus simple d'une liste est le point clé ou liste détaillée. Dans les classes KOMA-Script suivantes, l'énumération pour initialiser un élément de liste utilisé est « • », « – », « \* » et « · ».

La définition de caractère pour chaque niveau est stockée dans les macros `\labelitemi`, `\labelitemii`, `\labelitemiii` et `\labelitemiv` que vous pouvez facilement redéfinir avec `\renewcommand`.

```
\begin{itemize}
  \item...
  :
\end{itemize}
\labelitemi
\labelitemii
\labelitemiii
\labelitemiv
```

Chaque article est présenté avec `\item`.

Pour présenter une liste simple imbriquée dans plusieurs niveaux, vous écrirez :

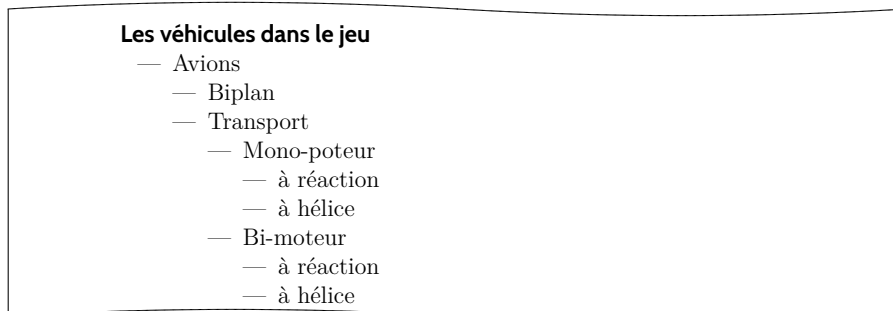
```
\minisec{Les véhicules dans le jeu}
\begin{itemize}
  \item Avions
    \begin{itemize}
      \item Biplan
        \item Transport
          \begin{itemize}
            \item Mono-moteur
              \begin{itemize}
                \item{à réaction}
                \item{à hélice}
              \end{itemize}
            \item Bi-moteur
              \begin{itemize}
                \item{à réaction}
                \item{à hélice}
              \end{itemize}
          \end{itemize}
        \end{itemize}
      \end{itemize}
    \end{itemize}
  \end{itemize}
```

```

\end{itemize}
\end{itemize}
\end{itemize}
\item Motos
\end{itemize}

```

pour obtenir quelque chose dans ce style :



La liste numérotée est fréquemment utilisée à partir du noyau de  $\text{\LaTeX}$ . En fonction du niveau, la numérotation est réalisée de différentes façons : chiffres arabes, minuscules, petits chiffres romains et lettres majuscules.

```

\begin{enumerate}
  \item...
  :
\end{enumerate}
\theenumi
\theenumii
\theenumiii
\theenumiv
\labelenumi
\labelenumii
\labelenumiii
\labelenumiv

```

Le type de numérotation du second niveau et le format de sortie sont déterminés par les macros de  $\text{\labelenumi}$  jusque  $\text{\labelenumiv}$ . Alors que la lettre minuscule de deuxième niveau est suivie d'une parenthèse ronde, les valeurs de tous les autres niveaux sont suivies par un point. Chaque article est introduit avec  $\text{\item}$ .

Dans la liste les étiquettes sont marquées normalement avec `\label` et peuvent ensuite être référencées par `\ref`. Ci-dessous, le label des avions mono-moteur à réaction porte l'étiquette `{xmp : à réaction}` dont la valeur `\ref` est alors « 1(b)iA ».

On peut raccourcir l'exemple précédent en utilisant `enumerate` pour le détail :

- Les véhicules dans le jeu**

  1. Avions
    - a) Biplan
    - b) Transport
      - i. Mono-moteur
        - A. à réaction
        - B. à hélice
      - ii. Bi-moteur
  2. Motos
  3. Automobile
    - a) voiture de course
    - b) voiture de tourisme
    - c) camion

Une autre forme de liste est la liste de description qui sert principalement à la présentation de l'un des termes individuels spécifié dans un paramètre facultatif de `\item`.

```
\begin{description}
  \item{mot-clé}...
  :
\end{description}
```

La police utilisée pour mettre en évidence le mot-clé peut également être changée en utilisant les commandes `\setkomafont` et `\addtokomafont` des classes **KOMA-Script** (voir la section 2.6) pour l'étiquette `description` de l'élément (voir la table 2.2 page 45) à utiliser avant `\sffamily\bfseries`.

Vous voulez que les mots-clés soient émis uniquement en **gras**, dans la police par défaut au lieu de **sans sérif**, définissez la police en conséquences avec :

```
\setkomafont{descriptionlabel}{\normalfont
```

```
\bfseries}
```

Un exemple de sortie d'une liste de mots clés est une liste des styles de page dont le code source pourrait être :

```
\begin{description}
\item{\textbf{empty}} style de page sans entête ni pied de page,
\item{\textbf{plain}} style de page sans titre de colonne,
\item{\textbf{headings}} style de page avec titre automatique,
\item{\textbf{myheadings}} style de page avec titre manuel.
\end{description}
```

Ce qui donne :

**empty** style de page sans entête ni pied de page,  
**plain** style de page sans titre de colonne,  
**headings** style de page avec titre automatique,  
**myheadings** style de page avec titre manuel.

Une autre forme de liste environnement de l'étiquetage est disponible uniquement pour les classes KOMA-Script, avec un modèle précédemment présenté qui peut être spécifié dans l'étiquetage, la profondeur de retrait est déterminée par l'indentation de tous les éléments.

```
\begin{description}[séparateur]{modèle le plus long}
\item{mot-clé}...
:
\end{étiquetage}
```

Une option permet de régler la limite entre l'article et le texte descriptif. La police utilisée pour mettre en évidence les mots-clés peut être modifiée avec les commandes `\setkomafont` et `\addtokomafont` pour l'étiquette de marquage et pour le séparateur d'étiquetage (voir la table 2.2 page 45). Nous pouvons décrire un environnement qui ressemble à :

```
\setkomafont{labelinglabel}{\ttfamily}
\setkomafont{labelingseparator}{\normalfont}
\begin{labeling}[~--]{ma rubrique}
\item[empty] est le style de page vide sans entête ni pied,
\item[plain] est le style pour chapitre sans titre,
\item[headings] est le style avec titre automatique,
```

`\item[myheadings]` est le style avec titre manuel.  
`\end{labeling}`

et nous obtenons :

<code>empty</code>	– est le style de page vide sans entête ni pied,
<code>plain</code>	– est le style pour chapitre sans titre,
<code>headings</code>	– est le style avec titre automatique,
<code>myheadings</code>	– est le style avec titre manuel.

Comme on peut le voir dans l'exemple, un changement de police est facilement réalisé en s'appuyant sur le mot-clé mais il affecte le séparateur et il peut être nécessaire de neutraliser la fonction de commutation. Cet environnement a été initialement développé pour des structures comme « où, recherche, solution », les plus fréquentes dans les notes de cours, alors qu'il peut être placé dans des applications très différentes. Ainsi, les exemples de ce manuel ont été définis avec l'environnement `labeling`. L'environnement `verse` n'est pas perçu comme un environnement de liste parce qu'il n'utilise pas `\item`, mais `flushleft` avec saut de ligne fixe. `\cott{begin}\{verse\} bla, bla, bla\dots \cott{end}\{verse\}`

Il est en interne dans les deux classes `standard` et `list` de KOMA-Script.

Utilisé pour les poèmes, les lignes sont tracées à gauche et à droite, les vers simples sont terminés par un saut de ligne fixe avec `\\` et les vers sont définis comme un paragraphe, séparés par une ligne vide. Souvent, on peut utiliser à la place `\medskip` ou `\bigskip`. Pour éviter la création d'une nouvelle page, utiliser `\\*` au lieu de `\\`. En exemple, un court poème de Wilhelm BUSCH, avec le résultat dans la version originale :

```
\begin{verse}
Wenn einer, der mit Mühe kaum\\*
Gekrochen ist auf einen Baum,\\*
Schon meint, dass er ein Vogel wär\\*
So irrt sich der.
\end{verse}
```



```

Wenn einer, der mit Mühe kaum
Gekrochen ist auf einen Baum,
Schon meint, dass er ein Vogel wär
So irrt sich der.

```

et la traduction <sup>5</sup>.

Dans un très long vers comme :

```

\begin{verse}
Le philosophe dit que le propriétaire a toujours des
réparations\\* \bigskip
Qui vous dit qu'il n'a jamais menti? \\
\hfill ne vous y fiez pas, mon fils!
\end{verse}

```

comportant un saut de ligne dans le texte :

```

Le philosophe dit que le propriétaire a toujours des ré-
parations

Qui vous dit qu'il n'a jamais menti?
ne vous y fiez pas, mon fils!

```

Le `\\*` n'est pas seulement un saut de ligne, il évite également un saut de page. Pour parvenir à cette fin, insérer à l'intérieur de la première ligne `\nopagebreak` :

```

\begin{verse}
Le philosophe, comme le propriétaire \nopagebreak{}
a toujours des réparations. \\
\bigskip
Qui vous dit qu'il n'a encore jamais \nopagebreak{}
menti, ne vous y fiez pas, mon fils.
\end{verse}

```

Voici ce dont vous devriez toujours tenir compte si vous êtes confrontés à des questions ou à des réponses apparemment étranges sur  $\text{\LaTeX}$  :

---

5. Si celui qui a, avec peine,  
rampé sur un arbre  
croit déjà être un oiseau,  
alors, il se trompe.

```
\begin{verse}
Nous n'aimons pas faire plaisir \\*
à qui peut faire, ce que nous ne pouvons pas.\\
\bigskip
Comme il parait petit celui de qui \\*
l'on a la mesure de sa vanité.
\end{verse}
```

Dans ces exemples, la façon dont chaque `\bigskip` est utilisé pour séparer, respectivement, les deux énonciations l'une de l'autre.

Nous n'aimons pas faire plaisir  
à qui peut faire, ce que nous ne pouvons pas.

Comme il parait petit celui de qui  
l'on a la mesure de sa vanité.

Ces deux environnements, `quote` et `quotation` se trouvent à la fois dans les classes `standard` et dans `KOMA-Script`.

```
\begin{quote} ... \end{quote}
\begin{quotation} ... \end{quotation}
```

Ils sont fréquemment utilisés pour déposer plus de citations avec le texte justifié en retrait à droite et à gauche. La différence entre les deux réside dans la manière dont les paragraphes sont composés, bien que définie par la distance verticale de citation en citation avec indentation horizontale de la première ligne. Ceci s'applique également à la première ligne d'un environnement d'énumération, sauf si `\noindent` est préfixé.

Vous voulez mettre en évidence une anecdote. Vous écrivez la citation en utilisant l'environnement :

```
\documentclass[paper=a5,\texttt{pagesize}]{scrartcl}
\usepackage{selinput}
\SelectInputMappings{adieresis={ä},germandbls={ß}}
\usepackage[ngerman, french]{babel}
\begin{document}
```

Un exemple sous la forme d'une anecdote qui, dit-on, aurait eu lieu autrefois en Souabe :

```
\begin{quotation}
```

On sonne à la porte d'une maison paroissiale de Stuttgart. La femme de ménage ouvre, un homme mal rasé se tient devant la porte, vêtu d'un manteau miteux, tenant un chapeau tricoté à la main.

-- Ma bonne dame, dit l'homme d'un ton douloureux, dans l'allemand le plus pur, je n'ai rien mangé depuis trois jours.  
La femme secoue la tête avec bienveillance et lui répond, avec la plus grande conviction, et en patois :

-- Mon dieu, mais vous devriez vous forcer!  
`\end{quotation}`

`\end{document}`

Le résultat pourrait ressembler à ceci :

Un exemple sous la forme d'une anecdote qui, dit-on, aurait eu lieu autrefois en Souabe :

On sonne à la porte d'une maison paroissiale de Stuttgart. La femme de ménage ouvre, un homme mal rasé se tient devant la porte, vêtu d'un manteau miteux, tenant un chapeau tricoté à la main.

– Ma bonne dame, dit l'homme d'un ton douloureux, dans l'allemand le plus pur, je n'ai rien mangé depuis trois jours.

La femme secoue la tête avec bienveillance et lui répond, avec la plus grande conviction, et en patois :

– Mon dieu, mais vous devriez vous forcer !

Je ne peux m'empêcher de citer, pour le plaisir, le texte « Schwäbisch »<sup>6</sup> original avec toute la saveur de la chute.

Ein kleines Exemple für eine Anekdote, die sich einst in Schwaben zugetragen haben soll:

Es klingelt an der Tür eines Pfarrhauses in Stuttgart. Als die Haushälterin öffnet, steht ein unrasierter Mann in reichlich

6. PAPIRAY précise : Schwäbisch = souabe, nom de la région sud-est de l'Allemagne comprise entre le Rhin qui la sépare à l'ouest de l'Alsace et au sud de la Suisse, le Lech qui la sépare à l'est de la Bavière, frontalière au nord de la Franconie qui occupe le bassin du Main, et dont le parler, ce fameux « Schwäbisch » ressemble furieusement à celui de leurs voisins alsaciens, bien que ces derniers s'en défendent avec véhémence.

schäbigem Mantel vor der Tür und hält seine Strickmütze in der Hand.

„Gute Frau“, verkündet der Mann in gequältem Ton, doch bestem Hochdeutsch, „ich habe seit drei Tagen nichts mehr gegessen“.

Die Frau schüttelt mitleidig den Kopf und entgegnet im Brustton vollster Überzeugung: „Guda Moh, Sie missat sich halt zwinga!“<sup>7</sup>

À titre de comparaison, l'ensemble est également représenté avec un environnement de citation, `quote` :

Un exemple sous la forme d'une anecdote qui, dit-on, aurait eu lieu autrefois en Souabe :

On sonne à la porte d'une maison paroissiale de Stuttgart. La femme de ménage ouvre, un homme mal rasé se tient devant la porte, vêtu d'un manteau miteux, tenant un chapeau tricoté à la main.

– Ma bonne dame, dit l'homme d'un ton douloureux, dans l'allemand le plus pur, je n'ai rien mangé depuis trois jours.

La femme secoue la tête avec bienveillance et lui répond, avec la plus grande conviction, et en patois :

– Mon dieu, mais vous devriez vous forcer !

```
\begin{addmargin}[retrait à gauche]{retrait}...
\end{addmargin}
\begin{addmargin}[indentation interne]{retrait}...
\end{addmargin}
```

Dans les contextes `quote` et `quotation`, `addmargin` modifie la limite de bordure mais ne change ni l'indentation ni l'espacement de paragraphe. Il n'existe pas d'espacement vertical inséré après cet environnement. Avec un argument obligatoire, l'indentation, donné, le contenu est en retrait vers la droite et vers la gauche de ce montant. Si l'argument optionnel, par exemple *retrait gauche* est donné seul, l'indentation gauche est laissée malgré l'entrée de la valeur ajoutée. La version étoilée `addmargin*` est

---

7. PAPIRAY précise : ce texte en allemand, encadré par les commandes `\begin{de}` et `\end{de}` respecte la typographie de la langue utilisée (merci babel).

différente de la variante sans étoiles, uniquement, dans l'ensemble recto-verso, lorsque l'argument indentation interne optionnel est utilisé. Dans ce cas, la valeur de l'empreinte interne est ajoutée à la partie du bord intérieur de la page, côté gauche de la page sur le côté gauche et côté droit sur le côté droite, l'aménagement est valable pour chaque face. Dans les deux versions de l'environnement, des valeurs négatives sont autorisées pour tous les paramètres, ce qui a pour effet de décaler l'environnement dans la marge.

Supposons que vous écrivez un manuel avec des exemples courts de code source. Pour mettre le texte en évidence, il sera encadré par des lignes horizontales faciles à déplacer vers le bord extérieur. Elles sont utilisées pour démontrer l'intérêt de la définition d'un environnement, défini lui-même à l'intérieur de cet environnement :

---

```

\newenvironment{cadre code source}{%
\begin{addmargin*}[1em]{-1em}%
\begin{minipage}{\linewidth}%
\rule{0.95\linewidth}{2pt}%
}{%
\rule[.25\baselineskip]{\linewidth}{2pt}%
\end{minipage}%
\end{addmargin*}%
}

```

---

Le résultat n'est pas nécessairement beau mais il décrit l'utilité qui peut en être faite.

L'argument optionnel de `addmargin*` veille à ce que le bord interne soit augmenté de 1em. À son tour le bord extérieur augmente d'une valeur négative, diminué dans la réalité de 1em, ce qui se traduit par un décalage de 1em vers l'extérieur. Bien entendu, au lieu de 1em, une longueur peut être fixée, par exemple à 2, en utilisant `\parindent`.

Une page à gauche ou à droite ne peut être détectée de manière fiable lors de la première exécution de L<sup>A</sup>T<sub>E</sub>X, voir à ce propos les explications des instructions `\ifthispageodd` (section 2.11) et `\ifthispagewasodd` dans la section 20.1. Dans l'interaction des environnements, telles les listes de paragraphes, des questions sont posées fréquemment. On trouvera dans la

même section des explications consacrées à l’option `parskip`, ainsi que le traitement de problèmes liés à la multi-pages `addmargin*`.

Et pour le plaisir, je ne résiste pas à la transcription du « dit » ci-après qui a pu figurer dans les brèves de comptoir de Jean-Marie GOURIOT, distillées par Jean CARMET :

```
\rule{0.95\linewidth}{1pt}%
\begin{quote}
\parindent=0pt
\begin{minipage}{0.9\textwidth} \small
Parfois, quand vous pleurez, personne ne remarque vos larmes.
Parfois, quand vous êtes triste, personne ne remarque votre
peine.
Parfois, quand vous êtes heureux, personne ne remarque votre
sourire.

Mais pétez juste une fois \dots
\end{minipage}
\end{quote}
\rule{0.95\linewidth}{2pt}%
```

Ce code produit :

---

Parfois, quand vous pleurez, personne ne remarque vos larmes.  
Parfois, quand vous êtes triste, personne ne remarque votre peine.  
Parfois, quand vous êtes heureux, personne ne remarque votre sourire.  
Mais pétez juste une fois...

---

## 2.19 Mathématiques

Les classes de KOMA-Script ne fournissent pas d’environnements pour les formules mathématiques, les équations ou les éléments similaires de mathématiques. La base KOMA-Script repose, dans ce cas, entièrement sur le noyau de L<sup>A</sup>T<sub>E</sub>X. Ceci s’applique également aux deux options `leqno` et `fleqn`.

Ici, pas de description des environnements de mathématiques du noyau  $\text{\LaTeX}$  et si vous souhaitez utiliser `displaymath`, `equation`, et `eqnarray` vous devriez lire une courte introduction  $\text{\LaTeX}$  telle que [DGS+12] ou [OPHS11]. Mais si vous voulez plus de mathématiques, l'utilisation du paq `amsmath` est recommandée (voir [Ame02]).

#### leqno

Normalement, les équations sont numérotées sur le côté droit et l'option `leqno` provoque le chargement du fichier d'options standard `leqno.clo` qui permet de les numéroter sur le côté gauche. Cette option doit être utilisée comme un argument facultatif de `\documentclass`. Son utilisation par `\KOMAOPTIONS` ou `\KOMAoption` n'est pas prise en charge. Ce n'est pas illogique puisque le paq mathématiques `amsmath`, qui la soutient lors de son chargement, ne réagirait pas sur les changements d'exécution de cette option.

#### fleqn

Les équations sont souvent délivrées centrées horizontalement et l'option standard `fleqn` provoque le chargement du fichier d'options `fleqn.clo`. Les équations sont ensuite justifiées à gauche. Cette option est utilisée comme argument de `\documentclass` mais pas par `\KOMAOPTIONS` ou par `\KOMAoption`. Comme pour `leqno`, et pour les mêmes raisons, l'utilisation par `\KOMAOPTIONS` ou `\KOMAoption` n'est pas prise en charge.

## 2.20 Environnement flottant pour figure et table

$\text{\TeX}$  fournit des environnements flottants, mécanisme très puissant et confortable pour le placement automatique des figures et tables.

Strictement parlant, on devrait utiliser « tableau » à la place de « table », ce qui permettrait de différencier les environnements `table` et `tabular`, et ce n'est pas évident en langue allemande qui, apparemment, utilise indifféremment le mot « Tabelle » (table). Souvent, le novice hésite sur la manière de définir la position du flottant, table ou figure, dans le texte. Or, cette position est fixée par une référence, sur la page si l'espace est suffisant ou, dans le cas contraire, sur la page suivante (ce qui crée souvent un espace vide inesthétique). C'est souvent le même argument optionnel

qui est utilisé pour positionner un objet dans un document, ce qui n'a aucun sens. Dans un tel cas, il est préférable de changer la valeur par défaut dans le préambule.

Une note importante avant de commencer cette section du préfixe : la plupart des mécanismes présentés ici vont au-delà des capacités des classes `standard` et ne fonctionnent plus lorsque vous utilisez un paq engagé dans la production des titres de table et figure et changent leur apparence. Cela devrait être évident, mais est souvent négligé, car oublié.

`captions=réglage`

Dans les classes normalisées, le texte de la légende est inséré par la commande `\caption` en dessous de la table ou de la figure, correct avec les figures, douteux avec les tables lorsque la légende doit être placée au-dessus lorsqu'elle couvre plusieurs pages. En outre, la table est lue ligne par ligne de haut en bas. C'est la raison pour laquelle `KOMA-Script`, contrairement aux classes `standard`, propose `\captionabove` pour les légendes au-dessus et `\captionbelow` pour les légendes au-dessous des tables et des figures, en plus du réglage classique `captions=tableheading`. Il convient de noter à ce stade que les tables multi-pages ne peuvent être définies comme flottants et que des paqs `add-on` sont requis pour la pagination automatique de tables telles que `longtable` (voir [Car04]) ou `tabu` (voir [Che11]). Avec les légendes `captions=tablesignature`, le formatage est sélectionné par défaut comme signature de la table. Il est à noter ici que les deux valeurs ne changent que la mise en forme. L'endroit où la légende est réglée, au-dessus ou au-dessous du flottant, dépend de `KOMA-Script` et du lieu où la commande `\caption` est utilisée. Toutefois, cela change lorsque vous utilisez le paq `float` avec la commande `\restylefloat` (voir [Lin01]).

Il existe aussi les fonctions correspondantes pour les photos, telles que les options `captions=figureheading` et `captions=figuresignature`. Dans de très rares cas, elles sont susceptibles d'être utiles pour modifier cette rubrique lors d'un formatage des légendes de figures et parfois, il faut aussi fournir tous les environnements avec des titres. D'autres options, `captions=heading` et `captions=signature`, permettent de modifier la mise en forme de tous les flottants en conformité avec `KOMA-Script`. Elles exercent leurs effets, même utilisées dans l'environnement `float`. Noter que, si vous utilisez le paq `float`, les paramètres des positions au-dessous



et au-dessus ne fonctionnent pas lorsque vous appliquez `\restylefloat` pour les tables ou les tableaux. Pour plus de détails, référez-vous à [Lin01]. Le soutien fourni en utilisant le paq `float` se trouve dans `komaabove`, les légendes sont réglées par `captions=nooneline` avec la distinction entre simples et multi-lignes, en les activant ou les désactivant. Dans les titres et les sous-titres d'une ligne, `caption=oneline` gère la légende centrée ou non (le réglage est *centré horizontalement* par défaut). Une particularité de `KOMA-Script`, dans la gestion des flottants, est qu'il est possible de placer le titre au-dessus ou au-dessous avec un contenu à teneur réelle. L'usage de l'environnement `captionsbeside` fait l'objet d'une explication particulière dans cet ouvrage, ses paramètres peuvent être modifiés avec l'option `captions`.

Les valeurs possibles figurent dans la table 2.17.

TABLE 2.17 – Valeurs possibles de mise en forme pour légendes de flottants

Éléments	Description
<code>bottombeside,</code> <code>besidebottom</code>	Titre de l'environnement <code>captionbeside</code> (voir la section 2.20) aligné avec la ligne de base du plus bas niveau de référence du contenu de l'environnement flottant.
<code>centeredbeside,</code> <code>besidecentered,</code> <code>middlebeside,</code> <code>besidemiddle</code>	Titre de l'environnement <code>captionbeside</code> centré verticalement.
<code>figureheading,</code> <code>figureabove,</code> <code>abovefigure,</code> <code>topatfigure</code>	En illustration, le formatage choisi est sélectionné comme titre (si nécessaire, peut différer de <code>captions=signature</code> ).
<code>figuresignature,</code> <code>figurebelow,</code> <code>belowfigure,</code> <code>bottomatfigure</code>	En illustration, est sélectionné comme légende de formatage (si nécessaire, peut différer de <code>captions=heading</code> ).

<code>heading, above, top</code>	Titre d'environnement flottant formaté comme les manchettes indépendant du fait qu'il soit placé au-dessus ou en dessous. L'option implique également <code>captions=tableheading</code> et <code>captions=figureheading</code> .
<code>innerbeside, besideinner</code>	Titre d'environnement flottant avec <code>captionbeside</code> (voir la section 2.20) à double côté fixé par défaut à l'intérieur du contenu de l'environnement flottant. Cela correspond à <code>captions=leftbeside</code> .
<code>leftbeside, besideleft</code>	Titre d'environnement flottant avec <code>captionbeside</code> (voir la section 2.20) fixé par défaut à gauche du contenu de l'environnement flottant.
<code>nooneline</code>	Manchette et sous-titre d'environnement flottant sur une seule ligne, ne sont pas traités séparément, mais comme multiligne.
<code>oneline</code>	Manchette d'une seule ligne et sous-titre flottant traités séparément à fin de centrage horizontal.
<code>outerbeside, besideouter</code>	Titre d'environnement flottant avec <code>captionbeside</code> (voir la section 2.20) à double côté fixé par défaut à l'extérieur en plus du contenu de l'environnement flottant. Cela correspond à un côté <code>captions=rightbeside</code> .
<code>rightbeside, besideright</code>	Titre d'environnement flottant avec <code>captionbeside</code> (voir la section 2.20) fixé par défaut à la droite du contenu de l'environnement flottant.
<code>signature, below, bot, bottom</code>	Titre d'environnement flottant formaté comme une légende. Toutefois, cela n'a aucune incidence sur le fait qu'il soit placé au-dessus ou en dessous de l'environnement flottant. L'option implique également <code>captions=tablesignature</code> et <code>captions=figuresignature</code> .

<code>tableheading,</code>	Pour la mise en forme du titre des tables
<code>tableabove,</code>	sélectionnées (si nécessaire, peut différer de
<code>abovetable,</code>	<code>captions=signature</code> ),
<code>abovetabular,</code>	
<code>topatable</code>	
<code>tablesignature,</code>	Pour la mise en forme de la légende des tables
<code>belowtable,</code>	sélectionnées (peut aussi différer),
<code>belowtabular,</code>	
<code>bottomatable</code>	
<code>topbeside, beside top</code>	Titre d'environnement flottant avec <code>captionbeside</code> aligné avec la ligne de base du plus haut niveau de référence du contenu de l'environnement flottant (voir la section 2.20).

<pre> \caption[<i>entrée de répertoire</i>]{<i>titre</i>} \captionbelow[<i>entrée de répertoire</i>]{<i>titre</i>} \captionabove[<i>entrée de répertoire</i>]{<i>titre</i>} </pre>
--

Le titre des tables et figures, dans les classes `standard`, est légendé en dessous en utilisant la déclaration `\caption`. La numérotation basique est correcte. Pour les tables, le titre est placé au-dessus ou en dessous.

Par conséquent `KOMA-Script` utilise `\captionabove` pour le nom placé avant ou au-dessus de l'objet à désigner et `\captionbelow` pour le nom placé sous ou à côté de ce même objet. Utiliser les options décrites dans cette section permet d'en modifier le comportement avec les tables et les figures ou d'autres environnements flottants et, pour des raisons de compatibilité le comportement par défaut de `\caption` dans ce cas est similaire à `\captionbelow`. Néanmoins, il est recommandé d'utiliser les titres de tables et à cet effet de changer le comportement de `\caption` à l'intérieur des environnements de tables en utilisant `\captionbelow` à l'aide de `captions=tableheading`. Sinon, utiliser `\captionabove` dans l'environnement de chaque table.

Vous voulez par exemple travailler avec des entêtes à la place des légendes de tables, parce que vos tables s'étalent sur plus d'une page :

```
\begin{table}
```

```

\caption{\label{t0218}ceci est juste un exemple de table}
\begin{tabular}{llll}
Ceci & est & un & exemple. \\ \hline
s'il vous & plaît & laissez & le \\
contenu & de cette & table & inaperçu.
\end{tabular}
\end{table}

```

TABLE 2.18 – Ceci est juste un exemple de table

Ceci	est	un	exemple
s'il vous	plaît	laissez	le
contenu	de cette	table	inaperçu

Vous obtenez un résultat qui ne vous satisfait pas.

La police pour la description et l'étiquette figure ou table suivie d'un nombre, ou avec un séparateur, peut être modifiée avec `\addtokomafont` et `\setkomafont` (voir la section 2.6). Les responsables de la légende des éléments sont ici `caption` et `captionlabel` (voir la table 2.2 page 45). La police de l'élément `caption` est appliquée initialement sur l'élément `captionlabel` avant l'utilisation de la police spécifique.

Les réglages par défaut sont présentés dans la table 2.17 page 123.

Avec KOMA-Script, pour obtenir un meilleur résultat, vous écrivez à la place :

```

\begin{table}[!h]
\centering
\captionabove{ceci est juste un exemple de table}
\setlength{\dashlinedash}{10pt} % longueur du trait pointillé
\setlength{\dashlinegap}{2pt} % espace entre deux pointillés
\renewcommand{\arraystretch}{1.3} % espacement du texte
\begin{tabular}{l:l:l:l} % ajout des traits verticaux
Ceci & est & un & exemple \\ \hline
s'il vous & plaît & laissez & le \\
contenu & de cette & table & inaperçu.
\end{tabular}
\end{table}

```

Vous obtenez le résultat souhaité :

TABLE 2.19 – ceci est juste un exemple de table

Ceci	est	un	exemple
s'il vous	plaît	laissez	le
contenu	de cette	table	inaperçu

Exemple : vous voulez une table et des tableaux avec légendes en petite police. Vous écrivez dans le préambule de votre document :

```
\addtokomafont{caption\index{\co{caption}}}{\small}
```

En outre, vous souhaitez que l'étiquette soit imprimée dans une police sans empattement et en gras. Vous ajoutez :

```
\setkomafont{captionlabel}{\sffamily\bfseries}
```

Le résultat serait, dans ce cas, le même qu'en utilisant `\addtokomafont`.

```
\captionof{type d'objet}[répertoire fiche]{titre}
\captionaboveof{type d'objet}[répertoire fiche]{titre}
\captionbelowof{type d'objet}[répertoire fiche]{titre}
```

D'une manière similaires, les paqs `caption` et `capt-of` de KOMA-Script proposent la commande `\captionof` avec laquelle vous définissez un titre approprié à l'entrée dans le répertoire respectif hors d'un environnement flottant, ou dans un environnement flottant étranger dont la nature doit être spécifiée comme premier argument supplémentaire par opposition à `\caption`. KOMA-Script propose aussi les commandes `\captionaboveof` et `\captionbelowof` identiques à `captionabove` et `captionbelow`, mais avec les caractéristiques et les paramètres supplémentaires de `\captionof`.

Bien entendu, `captionof` prend en compte l'environnement de l'option `caption`, les paramètres relatifs à la mise en forme du titre « sur » ou de la légende « sous » la table ou la figure. Mais cette capacité risque finalement d'être perdue lors du chargement de paqs tels que `capt-of` ou `caption`.

Un bon conseil : n'hésitez pas à consulter le tutoriel du paq `caption`.

Exemple : vous voulez créer un objet flottant, dans lequel une table et un graphique sont adjacents. Comme les flottants mixtes n'existent pas, vous utiliserez un environnement `figure` :

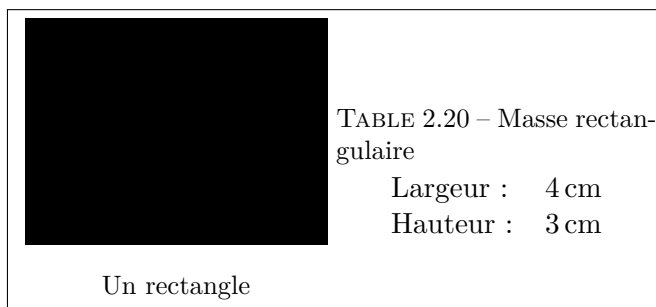
```

\begin{figure}
\begin{minipage}{.5\linewidth}
\centering
\rule{4cm}{3cm}
\caption{\label{t0220}un rectangle}\label{fig : rectangle}
\centering
\captionaboveof{table}
[dimensions du rectangle de la figure
\ref{fig :rectangle}]%
{dimensions du rectangle}
\label{tab : rectangle}
\begin{tabular}{l}
largeur : & 4\,cm \\
hauteur : & 3\,cm
\end{tabular}
\end{minipage}
\end{figure}

```

Pour définir la figure et la table côte à côte, deux environnements minipage sont utilisés et le signe % après le premier est important car sans lui, une distance supplémentaire serait fixée entre-eux. La légende de la figure est réglée avec `\caption`. Pour l'écriture de la table `\captionaboveof` a été utilisé. Le premier argument est donné dans la table et c'est pour cette raison que, malgré l'environnement figure, une légende de la table doit être créée.

FIGURE 2.2 – Utilisation de `\captionaboveof` dans un environnement flottant



L'entrée dans le répertoire des tables est défini, dans `\captionaboveof` par un argument optionnel. Sans la spécification de cet argument, c'est le titre du dernier argument utilisé qui serait inscrit dans le répertoire et bien

que le titre d'un objet flottant soit autonome, ce serait sans importance dans le répertoire de la table.

Par conséquent, un autre titre est utilisé ici avec l'argument optionnel `\captionabove` pour le répertoire. Le résultat en est représenté dans la figure 2.3 page 131.

De la même manière, comme indiqué dans l'exemple ci-dessus, une table peut avoir un titre dans un environnement figure et vous pouvez également définir une table fixe en dehors de tout environnement flottant. Ici aussi, une minipage devrait normalement être utilisée pour empêcher un saut de page entre le titre et la table.

Un environnement `flushleft` supplémentaire autour de la mini-page permet de réserver une distance agréable au-dessus et au-dessous et évite le retrait de paragraphe de cet environnement.

```
\begin{captionbeside}[annuaire]{titre de l'annuaire}
[placement]
[largeur][offset]
:
\end{captionbeside}
\begin{captionbeside}[annuaire]{titre de l'annuaire}
[placement]
:
\end{captionbeside}
```

Les titres peuvent figurer aussi à côté de la figure, en plus de la partie supérieure ou inférieure, Cette description s'inscrit normalement dans le bord inférieur de la figure. Vous pouvez, avec un peu d'habileté, utiliser deux déclarations `\parbox` comme dans les classes `standard`. Cependant, `KOMA-Script` offre un environnement spécial qui peut être utilisé dans le milieu flottant.

Le paramètre optionnel de la première entrée et le paramètre obligatoire de titre sont similaires aux mêmes paramètres de `\caption`, `\captionabove` ou `\captionbelow`. Le titre est placé à côté du contenu. Les paramètres qui déterminent cette mise en place à gauche ou à droite sont :

- l – à gauche (*links*),
- r – à droite (*rechts*),

`i` – à l’intérieur (*innen*) sur le côté droit de la gauche, sur le côté gauche de la droite,

`o` – à l’extérieur (*außen*) sur le côté droit de la droite, sur le côté gauche de la gauche.

La valeur par défaut est à droite, à côté du contenu de l’environnement. Cependant, ce paramètre peut être changé avec l’option `captions` et ses valeurs `innerbeside`, `leftbeside`, `outerbeside` et `rightbeside`. Si vous utilisez « `i` » (interne) ou « `o` » (externe), deux compilations peuvent être nécessaires pour atteindre un alignement correct.

Normalement, le contenu de l’environnement et le titre remplissent toute la largeur disponible. Cependant, la possibilité de spécifier une largeur différente, pour le paramètre optionnel `largeur`, existe. Il peut aussi être supérieur à la largeur du corps de texte.

Si vous spécifiez une largeur, la largeur utilisée est centrée par rapport à la largeur du corps de texte. Un argument optionnel par rapport au bord gauche peut être spécifié à la place pour compenser un décalage. Une valeur positive correspond à un décalage vers la droite, une valeur négative à un déplacement vers la gauche. Avec un décalage de `0pt`, la sortie est justifiée à gauche.

Le paramètre optionnel avec ajout d’une « étoile » permet un décalage, dans l’impression recto-verso, sur le côté gauche avec déplacement par rapport au bord droit. Une valeur positive correspond à un décalage vers l’extérieur, et une valeur négative à un décalage vers l’intérieure, donc un décalage de `0pt` signifie l’alignement avec le bord intérieur. Cette variante nécessite, dans certaines circonstances, deux exécutions de `LATEX` pour atteindre le bon décalage. L’alignement vertical est fait par défaut, ce qui signifie que la ligne de base inférieure du contenu des flottants et la base inférieure de titres sont à un même niveau.

Les valeurs optionnelles `topbeside`, `centeredbeside` et `bottombeside` de `captions` permettent de modifier ce réglage.

Lors du réglage de `topbeside`, les lignes de la base supérieure du contenu du flottant et le titre sont alignés en hauteur, tandis que `centeredbeside` règle le centrage. Dans ce contexte, il convient de mentionner que les figures ont la référence généralement en dessous, ce qui peut être modifié, par exemple, avec la déclaration `raisebox`.




Exemple d'utilisation de l'environnement `captionbeside` formaté avec :

```
\begin{figure}
\begin{captionbeside}%
[Exemple: Description de la figure sur le côté]%
{Description de la figure qui est ni au-dessus, ni au-dessous
de l'illustration, mais sur le côté}%
[i][\linewidth][%
\dimexpr\marginparwidth+\marginparsep\relax]*
\fbbox{%
\parbox[b][5\baselineskip][c]{.95\textwidth}
{%
\hspace*{\fill}\KOMAScript
\hspace*{\fill}\par
}%
}
\end{captionbeside}
\label{captionbeside}
\end{figure}
```

La largeur totale est la largeur `\linewidth` disponible, mais l'ensemble est décalé vers l'extérieur avec `\marginparwidth + \marginparsep`. Le titre (ou la description) est placé à l'intérieur à côté de la figure. L'illustration elle-même est déplacée vers le haut et le bord de la colonne marginale.

FIGURE 2.3 – Description de la figure qui est ni au-dessus, ni au-dessous de l'illustration, mais sur le côté droit, alignée en bas.



La figure suivante montre le centrage de la description avec le code : `\KOMAoptions{captions=centeredbeside}`.

Enfin, l'alignement ci-après utilise `\KOMAoptions{captions=topbeside}`

Pour illustrer la façon dont vous pouvez utiliser `\raisebox` pour décaler la ligne de base, un exemple complet est donné ci-après. Ce changement, indiqué précédemment, peut être appliqué non seulement à un graphique

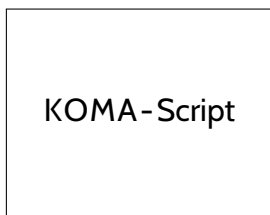


FIGURE 2.4 – Description de la figure qui est ni au-dessus, ni au-dessous de l’illustration, mais sur le côté droit, alignée au milieu.

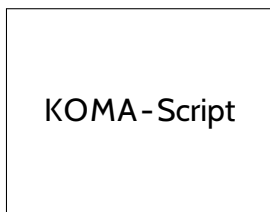


FIGURE 2.5 – Description de la figure qui est ni au-dessus, ni au-dessous de l’illustration, mais sur le côté droit, alignée en haut.

de remplacement, mais aussi, par exemple, avec `\includegraphics` (voir [Car05]) :

```
\documentclass[captions=topbeside]{scrbook}
\usepackage[french]{babel}
\usepackage{graphics}
\begin{document}
\chapter{Un exemple}
\begin{figure}
\begin{captionbeside}%
[Exemple:Description de la figure à côté, en haut]%
{Une description de la figure en haut, en plus de la figure du
paq \texttt{mwe}}%
[i][\linewidth][%
\dimexpr\marginparwidth+\marginparsep\relax
]*
\raisebox{%
\dimexpr\baselineskip-\totalheight\relax
}{%
\includegraphics{exemple-figure-1x1}%
}%
\end{captionbeside}
\label{fcaptionbesidetop}
```

```
\end{figure}
\end{document}
```

```
\begin{captionofbeside}{type d'objet}[répertoire]{titre}
[statut][largeur][offset]
:
\end{captionofbeside}
\begin{captionofbeside}{type d'objet}[répertoire]{titre}
[statut][largeur][offset]*
:
\end{captionofbeside}
```

Une variante `\captionof` existe, comme avec `\caption`, où le genre d'objet n'est pas déterminé par l'utilisation d'un environnement flottant à l'intérieur de ce type, dans le but de faire correspondre l'environnement `captionbeside` avec `captionofbeside`. L'objectif est aussi de préciser, contrairement à `captionbeside`, que le type d'objet peut désigner ici un environnement approprié comme premier argument supplémentaire.

```
komaabove
komabelow
```

Lorsque vous utilisez le paq `float` l'apparition des flottants définis de la sorte est déterminée uniquement par le style de flottant, ce qui nécessite de savoir si on doit travailler avec un titre ou une légende. Dans `float`, il n'existe pas de style prédéfini correspondant pour `KOMA-Script`, avec paramètres (voir ci-dessous) fournis. `KOMA-Script` utilise donc les deux styles `komaabove` et `komabelow`. Ceux-ci peuvent être activés, définis par les styles `plain`, `boxed` ou `ruled`, en utilisant le paq `float` (voir à ce sujet [**Lin01**]). Avec `komaabove`, `\caption`, `\captionabove` et `\captionbelow` le style est celui de titres alors qu'avec `komabelow` il est celui d'une légende sous l'illustration

```
\captionformat
```

Différentes options permettent de modifier la mise en forme, celles de la police ont déjà été expliquées. Le séparateur entre l'étiquette et le texte de l'étiquette est stocké par la macro `\captionformat`, nonobstant toutes les autres commandes de formatage, et donc, dans ce cas, ne contient pas de compteur mais uniquement des informations sur le compteur inclus.

La définition originale est :

```
\newcommand*\captionformat}{:}
```

ce qui peut être modifié avec `\renewcommand`.

Exemple : vous ne voulez pas, comme séparateur, deux-points suivis d'une espace mais suivis d'un tiret, y compris l'espace nécessaire. Par conséquent, vous redéfinissez `\captionformat` :

```
\renewcommand*\captionformat}{~--~}
```

Cette définition devrait figurer, par exemple, dans le préambule de votre document.

<pre>\figureformat \tableformat</pre>
---

Il a déjà été souligné que `\captionformat` ne contient pas de formatage pour l'étiquette elle-même. Ce devrait, maintenant, n'être qu'un chemin pour la redéfinition des commandes de sortie des compteurs `\thefigure` ou `\thetable`. Cette redéfinition peut avoir un impact sur répertoires ou `\ref`. Au lieu de cela, KOMA-Script propose deux commandes de type `\...format`.

Celles-ci sont prédéfinies comme suit :

```
\newcommand*\figureformat{\figurename~\thefigure\autodot}  
\newcommand*\tableformat{\tablename~\thetable\autodot}
```

Elles peuvent également être ajustées en utilisant `\renewcommand`.

Exemple : de temps en temps, il est souhaitable que les textes descriptifs soient délivrés sans aucune étiquette et bien sûr sans séparateur. Dans KOMA-Script les définitions suivantes sont suffisantes pour atteindre cet objectif :

```
\renewcommand*\figureformat{}  
\renewcommand*\tableformat{}  
\renewcommand*\captionformat{}
```

Noter toutefois qu'une nouvelle numérotation n'est pas émise, mais qu'elle continue à courir. Ceci est particulièrement important si les redéfinitions s'appliquent uniquement aux environnements particuliers figure ou table.

<pre> \setcapindent{collection} \setcapindent*{Xcollection} \setcaphanging </pre>
---

Comme déjà indiqué, les descriptions ne sont pas prédéfinies dans les classes `standard`. Cela signifie que, dans les descriptions multilignes, la deuxième ligne commence directement sous l'étiquette et que les classes `standard` ne contiennent aucun mécanisme d'influence directe. Toutes les lignes du texte qui ne sont plus à partir de cette deuxième ligne sous le label figure ou table commencent la première rangée sous la première ligne. Ce comportement correspond à l'utilisation de `\setcaphanging` et peut être modifié en utilisant la commande `\setcapindent` ou `\setcapindent*`.

Le paramètre d'entrée spécifie dans quelle mesure devrait être le retrait de la deuxième rangée et si, après l'étiquette et avant le texte de description doit encore figurer un saut de ligne en définissant l'indentation `Xentrée`. Au lieu de cela, utilisez la version étoilée de la commande `\setcapindent*`. En utilisant à la place et avant la description, une valeur négative de tiret, une rupture se produit et seule la première ligne de la description est en retrait, mais pas les suivantes, quel que soit le nombre des entrées. La description d'une ligne est sélectionnée avec les légendes de l'option, ainsi que la description multiligne fixée ou soumise à un traitement spécial, voir page 130 la déclaration des valeurs `online` et `noonline` de l'option.

Les figures 3.A à 3.D montrent les effets de différents paramètres. Il est clair que, avec une faible largeur de colonne, la disposition qui en dépend est défavorable. Le code source de la seconde figure est reproduit ici à titre d'exemple avec une signature mise à jour :

```

\begin{figure}
\setcapindent{1em}
\fbbox{\parbox{.95\linewidth}{%
\centering\KOMAScript}}
\caption{Exemples avec indentation de la deuxième rangée}
\end{figure}

```

Comme on peut le voir, la mise en forme peut donc également être modifiée localement dans l'environnement de figure. Le changement sera alors appliqué uniquement pour une figure. Les figures suivantes sont à

KOMA-Script

**Figure 2.6** – La valeur par défaut avec l'utilisation de `\setcaphanging`

KOMA-Script

**Figure 2.7** – Avec retrait négatif partiel et indentation à partir de la deuxième ligne en utilisant `\setcapindent{1em}`

KOMA-Script

**Figure 2.8** – Avec indentation de la deuxième rangée et mise en page avant la description en utilisant `\setcapindent*{1em}`

KOMA-Script

**Figure 2.9** – Avec la simple indentation de la deuxième rangée et une pause avant la description en utilisant `\setcapindent{-1em}`

nouveau avec les réglages de base ou les paramètres globaux que vous avez faits, par exemple, dans le préambule de l'ensemble du document .

Bien sûr, cela vaut même pour les tables.

```
\setcapwidth[orientation]{largeur}
\setcapmargin[orientation]{largeur}
\setcapmargin[bord gauche]{bord}
\setcapmargin*[bord intérieur]{bord}
```

L'utilisation de ces trois commandes peut influencer la disposition et la largeur de la description. Normalement, le texte entier et toute la largeur de la colonne sont disponibles.

Cette largeur peut être réduite avec la déclaration `\setcapwidth`.

L'argument obligatoire qui spécifie la largeur maximale utilisée pour la description. disponible comme un argument optionnel, peut être précédé d'une lettre précisant l'alignement horizontal exact de la description.

Les justifications possibles sont données dans la liste suivante :

- **l** aligné à gauche,
- **c** centré,
- **r** aligné à droite,
- **i** intérieur, aligné à droite sur le côté droit, aligné à gauche sur le côté gauche ;

- o extérieur, aligné à gauche sur le côté droit, aligné à droite sur le côté gauche.

L'orientation intérieure et extérieure correspond à un côté justifié à gauche et justifié à droite.

Dans `longtable`, les alignements intérieur et extérieur ne fonctionnent pas correctement.

En particulier dans ces tableaux, les pages suivantes sont alignées selon les descriptions de la première table de la pièce, ce qui est un problème conceptuel de `longtable`.

Avec la commande `\setcapmargin`, on peut définir la marge supérieure au lieu de la largeur de la description, avantage qui peut être spécifié et doit être respecté en plus de la marge de texte normal. Si vous voulez que les marges gauche et droite soient identiques, elles peuvent l'être ensemble au moyen d'un argument optionnel.

Avec la version étoilée `\setcapmargin*` va prendre le bord gauche dans la présentation double face intérieure définie autrement. Ici, dans les tables `longtable` se pose le même problème que dans la justification intérieure ou extérieure de la déclaration `\setcapwidth`. L'utilisation de `\setcapmargin` ou `\setcapmargin*` active en outre `caption=nooneline` (voir page 130) pour les descriptions de légendes qui sont composées avec ce paramètre de marge.

Vous pouvez également soumettre des valeurs négatives pour la marge et la marge de droite ou la marge extérieure. On obtient ainsi une description qui fait saillie sur le bord correspondant. Pour les experts et les utilisateurs expérimentés, l'application `\setcapwidth`, un peu plus délicate, peut être trouvée dans [Koh14a].

#### `origlongtable`

Si les titres de table du paq `longtable` (voir [Car04]) ne doivent pas être redéfinis par les classes `KOMA-Script`, l'option peut être réglée par `origlongtable`. Cette option doit être utilisée comme un argument optionnel de `\documentclass`. Un réglage via `\KOMAOPTIONS` ou `\KOMAoption` n'est pas pris en charge.

### `listof=réglage`

Normalement, les listes des flottants tels les tables et les figures, sont non numérotées ou ajoutées à la table des matières. Ce comportement peut être considéré, du point de vue des répertoires eux-mêmes, comme une alternative aux paramètres mentionnés `toc=nolistof`, `toc==listof` et `toc=listofnumbered`, et l'on peut obtenir les mêmes résultats avec les réglages `listof=notoc`, `listof=totoc` et `listof=numbered`.

Le plus haut niveau disponible de détails dans le cadre `\part` est utilisé pour les titres des répertoires, dans `scrbook` et `scrreprt` au niveau chapitre et dans `scrartcl` au niveau `section`. L'utilisation de `listof=leveldown` permet cependant l'usage d'un niveau hiérarchique inférieur.

Vous voulez, par exemple, mettre dans un livre des figures ainsi que le répertoire de tables et les sous-répertoires d'un répertoire commun, figures et tables. Vous utilisez simplement :

```
\KOMAoption{listof}{leveldown}
```

et ensuite à l'endroit approprié dans votre document :

```
\addchap*{figures et tables}
\listoffigures
\listoftables
```

Pour plus de détails sur la commande `\addchap*` consulter la `section 2.16`.

Normalement, les listes de flottants sont formatées à une largeur fixe est utilisées pour le nombre. Dans le même temps, toutes les entrées sont uniques, ce qui équivaut à l'utilisation du paramètre `listof=graduated`. Si les nombres deviennent trop grands parce que, par exemple, beaucoup de tables sont présentes, ou que l'endroit envisagé ne suffit pas, `toc=flat` sera utilisé pour la table des matières fournie par `KOMA-Script` donc pour la listes des flottants. La largeur des numéros est déterminée avec une espace en conséquence, automatiquement.

En ce qui concerne les effets secondaires, le fonctionnement, `toc=flat` applique ce qui a été expliqué dans la `section 3.9`.



Cependant, il est de nouveau souligné que la mise en forme avec `toc=flat` nécessite plusieurs exécutions de L<sup>A</sup>T<sub>E</sub>X avant que les listes ne prennent leur forme définitive.

Si le réglage `listof=entryprefix` est utilisé, la liste `listof=flat` est activée automatiquement.

Habituellement, envoyer chaque entrée dans une des listes de flottants avec un préfixe comme `figure` ou `table` n'a aucun sens. Il y a bien sûr, dans la table, uniquement des chiffres mais seulement pour trouver des figures dans la table des tableaux et un tel préfixe qui n'a pas de valeur supplémentaire est omis par défaut.

Avec le réglage `listof=entryprefix` un préfixe est à définir.

Toutes les entrées d'un répertoire auront le même préfixe qui dépend de la fixation du fichier d'aide utilisé pour ce répertoire.

Pour la table des figures ayant le fichier `lof` joint, `\listoflofentryname` est utilisé alors que pour la table de répertoire ayant le même fichier `lof`, ce sera `\listoflotentryname`.

Les classes `scrbook` et `scrreprt` ajoutent, au début de chaque chapitre K<sup>O</sup>M<sup>A</sup>-S<sup>C</sup>R<sup>I</sup>P<sup>T</sup>, une distance verticale par défaut dans les listes de flottants. Ce comportement, disponible également dans les classes `standard`, permet de regrouper les répertoires par chapitre.

Il correspond au réglage K<sup>O</sup>M<sup>A</sup>-S<sup>C</sup>R<sup>I</sup>P<sup>T</sup> `listof=chaptergap`. Ici, une distance verticale fixe de 10pt est utilisée.

Avec le réglage `listof=chaptergapline` on peut atteindre une distance verticale d'une ligne.

Avec `listof=nochaptergap` on peut désactiver, en totalité, la distance verticale.

La liste `listof=chapterentry` présente une caractéristique particulière. Ici, l'entrée de table des matières pour le chapitre dans la liste des flottants sera insérée à la place de la distance.

Noter qu'une telle entrée a lieu également lorsque le chapitre ne contient aucun environnement flottant. On peut influencer encore plus directement, sur ce qui devrait se produire avec de nouveaux chapitres dans les listes

d'environnements flottants, avec les options `chapteratlists` présentées dans la [section 2.16](#).

Un aperçu de toutes les valeurs possibles pour le réglage de `listof` peut être trouvé dans le tableau suivant.

TABLE 2.21 – Valeurs possibles de l'option `listof` pour ajuster forme et contenu des répertoires des flottants

Éléments	Description
<code>chapterentry,</code> <code>withchapterentry</code>	Les débuts de chapitre sont dans le répertoire des flottants à travers une entrée de table des matières dans le chapitre.
<code>chaptergapline,</code> <code>onelinechaptergap</code>	Les débuts de chapitre sont dans le répertoire des flottants à une distance marquée par une ligne.
<code>chaptergapsmall,</code> <code>smallchaptergap</code>	Les débuts de chapitre sont dans le répertoire des flottants à une petite distance mise en évidence.
<code>entryprefix</code>	Chaque entrée de répertoire est fournie, selon le préfixe de répertoire, avant le numéro. Le préfixe est généralement fonction de la langue, par exemple, dans les paramètres de langue française « figure » pour la table des figures et « table » pour la liste des tables, dans chaque cas suivies d'une espace.
<code>flat, left</code>	Les listes des flottants ont une forme de table avec numéro d'environnement dans la première colonne, titre dans la deuxième colonne et numéro de page dans la troisième colonne avec un espace pour le numéro du flottant dépendant de l'espace requis lors de la précédente exécution de <code>L<sup>A</sup>T<sub>E</sub>X</code> .
<code>graduated,</code> <code>indent,</code> <code>indented</code>	Listes de flottants sous une forme hiérarchique. L'espace disponible pour les numéros des environnement flottants est limité.

<code>indenttextentries,</code> <code>indentunnumbered,</code> <code>numberline</code>	La propriété des numéros de lignes (voir l'article 15.2) est réglée pour les listes des flottants, par exemple, les figures et la liste des tables. Ces entrées non numérotées sont alignées à gauche avec le texte de l'ensemble des entrées numérotées de même niveau. Cependant, les classes <code>KOMA-Script</code> elles-mêmes ne fournissent aucune entrée non numérotée dans ces répertoires. Cela affecte donc seulement les entrées qui ne sont pas générées, elles-mêmes, par les classes correspondantes, mais avec l'aide de <code>\addxcontentsline</code> (voir l'article 15.2).
<code>leftaligntextentries,</code> <code>leftalignunnumbered,</code> <code>nonumberline</code>	La propriété du numéro de ligne est par exemple pour la liste des flottants, le répertoire défini des figures et des tables. Ces entrées non numérotées sont alignées à gauche avec le numéro de l'ensemble des entrées numérotées de même niveau. Cependant, les classes <code>KOMA-Script</code> elles-mêmes ne fournissent aucune entrée non numérotée dans ces répertoires. Cela affecte donc seulement les entrées qui ne sont pas générées, elles-mêmes, par les classes correspondantes, mais avec l'aide de <code>\addxcontentsline</code> (voir l'article 15.2).
<code>leveldown</code>	Les répertoires sont déplacés à un niveau de détail vers le bas.
<code>nochaptergap,</code> <code>ignorechapter</code>	Les débuts de chapitre ne sont pas repérés dans les environnements flottants.
<code>notoc,</code> <code>plainheading</code>	Les listes de flottants, telles la liste des figures et la liste des tables se verront refuser l'entrée dans la table des matières.
<code>numbered,</code> <code>totocnumbered,</code> <code>tocnumbered,</code> <code>numberedtoc</code>	Les listes de flottants, telles la liste des figures et la liste des tables ont une entrée dans la table des matières et sont numérotées.

`totoc`, `toc`, `notnumbered` Les listes de flottants, telles la liste des figures et la liste des tables ont une entrée dans la table des matières sans être numérotées.

```
\listoftables
\listoffigures
```

Avec ces instructions, une liste de tables ou de figures peut être sortie. Les changements qui ont un impact sur ces répertoires deviennent visibles après deux compilations  $\LaTeX$ .

La disposition des listes peut être graduée par l'option `listof` affectée avec les valeurs `flat`.

En outre, les valeurs `listof` et `listofnumbered` agissent indirectement sur l'option `toc` (voir [section 2.9](#)), ainsi que les valeurs `totoc` et `totocnumbered` de l'option `listof` ci-dessus sur les répertoires.

Habituellement, vous trouverez les listes de flottants, tels les tables et figures, immédiatement après la table des matières.

Dans certains documents, elles sont même dans l'annexe, mais Markus KOHM, l'auteur, préfère l'emplacement en fonction de la table des matières.

### 2.21 Note de bordure

En plus de la zone de texte proprement dite, qui est normalement remplie dans l'espace d'impression, existent, dans les documents, des colonnes dites marginales dans lesquelles des notes (de marge) peuvent être placées. Dans ce document, je n'ai fait usage de ce procédé qu'à titre d'exemple, même à l'intégration du chapitre `scrjura`.

```
\marginpar[note de bordure gauche]{note}
\marginline{note}
```

Pour les notes de marge,  $\LaTeX$  utilise normalement la commande `\marginpar`. La note est placée dans la marge, sur le bord extérieur. Dans les documents recto, le bord droit est utilisé. Avec `\marginpar`, le côté peut éventuellement être modifié, avec la note dans la marge de gauche. Les notes de

marge sont toujours justifiées. L'expérience montre qu'il est souvent préférable de placer les notes de marge justifiées sur le bord gauche de l'empage plutôt que flottante à droite. Pour ce faire, `KOMA-Script` propose la commande `\marginline`.

`scrartcl`

Dans l'exemple de ce chapitre, la déclaration de classe doit être placée dans la marge, sur le bord comme ici, ce qui est réalisé avec la commande :

```
\marginline{\texttt{scrartcl}}
```

Au lieu de la commande `\marginline` l'utilisation de `\marginpar` aurait été possible. En fait, l'utilisation de `\marginline` ne fait rien de plus que :

```
\marginpar[\raggedleft\texttt{scrartcl}]
{\raggedright\texttt{scrartcl}}
```

Cette commande `\marginline` est donc juste une abréviation.

Pour les experts, des problèmes éventuels liés à l'utilisation de `\marginpar`, sont documentés et décrits dans la [section 21.1](#). Ils valent également pour `\marginline`. Par ailleurs, un paq `column` qui peut être réalisé avec sa propre pagination est présenté au [chapitre 19](#).

## 2.22 Annexe

L'annexe d'un document se compose des pièces jointes à ce document, c'est-à-dire des pièces typiques qui sont la bibliographie, l'index et un glossaire des termes. Ainsi, pour ces parties, vous ne devriez pas commencer à l'annexe, puisque ces pièces possèdent généralement déjà en elles, une distinction qui les rend reconnaissables en tant que pièces jointes.

Mais dans les notes figurent d'autres parties tels des documents étrangers cités, des notes de fin ou des tables, afin que les pièces mentionnées ci-dessus soient également énoncées dans l'annexe.

appendix

La déclaration `\appendix` introduit l'annexe dans les classes standard et les classes `KOMA-Script`.

Cette commande démarre, entre autres choses, la numérotation des chapitres, tout en contrôlant que les règles pour la numérotation des niveaux hiérarchiques par [DUD96] sont respectées. Ces règles sont expliquées plus en détail dans l'option description des numéros, à la section 2.16.

La forme des entêtes de chapitre de l'annexe est déterminée par les options `chapterprefix` et `appendixprefix`. Pour plus de détails, consulter la section 2.16.

Noter que cette annexe est une commande, pas un environnement avec `\`. La déclaration n'attend pas, non plus, un tel argument.

Les chapitres ou sections de l'annexe sont normalement fixés par `\chapter` et `\section`.

## 2.23 Bibliographie

La bibliographie ouvre des sources externes. Elle est, en règle générale, générée à l'aide du programme `BIBTEX` à partir d'un fichier ayant une structure de base de données comparables. Ceci peut être changé via le style `\bib`, à la fois dans la forme des objets ainsi que dans leur tri. Si, en plus, un paq littérature, par exemple `natbib`, `babel` ou `biblatex` est employé, l'influence de `KOMA-Script` s'estompe de la bibliographie et dans ce cas, les instructions du paq utilisé doivent être absolument respectées.

En ce qui concerne l'utilisation générale d'une bibliographie voir la référence [DGS+12].

bibliography=réglage

La configuration initiale peut être sélectionnée avec tout style défini de mise en forme. Deux de ces styles de formatage sont prédéfinies pour la bibliographie avec `KOMA-Script`. Toutefois, ceux-ci ne doivent pas être confondus avec les différents styles pour `bib`, dont l'un est sélectionné avec `\bibstyle`. Alors que `BIBTEX` détermine à la fois le type de tri et le contenu de la bibliographie, les paramètres de `KOMA-Script` peuvent seule-

ment affecter les caractéristiques de base de la bibliographie ou quelques propriétés de mise en forme des entrées.

Avec `bibliography=oldstyle`, un format `compact` exécuté dans les entrées individuelles par la déclaration `\newblock` avec une distance horizontale qui est extensible, est sélectionné. Le nom vient du fait que ce soit la forme la plus commune d'une bibliographie. En revanche, vous pouvez rejoindre une forme ouverte plus moderne avec le réglage `bibliography=openstyle`. Ici, le nom vient du fait que la commande `\newblock` insère un paragraphe. Les entrées dans la bibliographie sont ainsi plus divisées. Elles sont moins compactes et considérablement assouplies ou ouvertes. La possibilité de définir de nouveaux styles de mise en forme est décrite dans `\newbibstyle` section 20.3.

Outre le style de formatage, une autre propriété peut être aussi modifiée. La bibliographie est un type de répertoire dans lequel non seulement le contenu de la présente liste, mais aussi un contenu externe, est référencé. Cette réflexion permet de dire que la bibliographie est un chapitre ou une `section` séparée et mérite d'être numérotée. Le réglage `bibliography=totocnumbered` conduit à une entrée différée dans la table des matières. Je suis moi-même d'avis qu'avec un raisonnement similaire, on peut envisager qu'une liste classique de sources puisse créer un chapitre distinct. En outre, la bibliographie n'est pas écrite par vous-même.

Par conséquent, elle mérite au mieux, une entrée non numéroté dans la table des matières, ce qui est atteint avec le réglage `bibliography=totoc`. La valeur par défaut d'un chapitre, non numéroté dans la bibliographie, réglé sans entrée dans sa propre table des matières, correspond à `bibliography=nototoc`. Voir aussi l'option `toc` dans la section 2.9, en particulier la page des valeurs `bibliographynumbered`, `bibliography` et `nobibliography`.

Dans certains cas, tous les documents sont fournis avec une bibliographie unique, mais chaque chapitre d'un ensemble `scrbook` ou `scrreprt` obtient sa propre bibliographie et il est utile de vérifier si la bibliographie ne se situe pas sur le chapitre, mais quelque part, à un niveau plus profond de la `section`. Ceci peut être réalisé avec l'option `bibliography=leveldown` qui peut, par exemple, être également utilisée si la bibliographie doit apparaître avec d'autres répertoires sous un titre commun. Par conséquent, cette option est également disponible dans `scrartcl`. Un résumé des va-

leurs possibles pour le réglage peut être trouvé dans la table 2.23 page 149. Cependant, il est à noter que, avec `\newbibstyle`, des valeurs supplémentaires peuvent être définies.

```
\setbibpreamble{Préambule}
```

Avec la déclaration `\setbibpreamble` un préambule peut être réglé pour la bibliographie. La condition est qu'il le soit avant la commande pour activer la bibliographie. Cela ne doit pas être au même moment mais, par exemple, au début du document.

Comme l'option `bibliography=totoc` ou `bibliography=totocnumbered`, la déclaration ne peut être couronnée de succès, à moins qu'un paq ne soit chargé, qui redéfinisse l'environnement `thebibliography`.

Bien que le paq `natbib` n'utilise pas les macros internes de `KOMA-Script`, l'objectif pourraient être atteint par `\setbibpreamble` avec la version actuelle de `natbib` (voir [Dal10]).

Par exemple, vous voulez que la bibliographie ne soit pas, dans le document, dans l'ordre de citation mais dans l'ordre alphabétique. Utiliser la déclaration suivante :

```
\codd{setbibpreamble}{les références sont classées par
ordre alphabétique selon les noms des auteurs. Si
plusieurs auteurs existent, elles seront triées
d'après le premier auteur. \codd{par}\codd{bigskip}}
```

La commande `\bigskip` permet de régler l'espace entre préambule et première référence.

```
\setbibpreamble{préambule}
```

Avec la déclaration `\setbibpreamble`, un préambule peut être réglé pour la bibliographie. La condition est que l'instruction en soit émise avant de créer la bibliographie, par exemple au début du document, avec des options telles que `bibliography=totoc` ou `bibliography=totocnumbered` à la condition qu'un paquet qui empêche de redéfinir l'environnement `thebibliography=` ne soit pas chargé.

Bien que le paq `natbib` utilisé par les macros internes de `KOMA-Script` ne soit pas inédit, des travaux peuvent être réalisés avec `\setbibpreamble` qui fonctionne avec la version actuelle (voir [Dal10]).



TABLE 2.22 – Valeurs prédéfinies pour régler la forme de l’option bibliographie

<code>leveldown</code>	La bibliographie sera décalée d’un niveau hiérarchique vers le bas.
<code>nottotoc</code>	La bibliographie ne reçoit aucune entrée dans la table des matières et n’est pas numérotée.
<code>oldstyle</code>	Le format compact classique est choisi, avec <code>\newblock</code> dont seul la distance horizontale est extensible.
<code>openstyle</code>	Un format ouvert subdivisé est choisi, illustrant un paragraphe <code>\newblock</code> .
<code>totoc</code>	La bibliographie aura une entrée dans la table des matières, sans être numérotée.
<code>totocnumbered</code>	La bibliographie aura une entrée dans la table des matières et sera numérotée.

`\BreakBibliography{interruption}`

La déclaration existe uniquement lorsque l’environnement `thebibliography=` n’est pas défini par un nouveau paq. Dans ce cas, il est possible d’interrompre avec la bibliographie. L’interruption est alors sortie à l’intérieur d’un groupe. Cette Interruption pourrait par exemple être un titre en utilisant `\minisec`.

Malheureusement, il n’est pas possible de produire cette instruction au moyen d’une entrée spécifique dans la base de données `BBTEX`. Par conséquent, elle ne peut, actuellement, bénéficier qu’aux utilisateurs qui modifient eux-même la bibliographie, ce qui limite l’avantage de son usage.

```
\AfterBibliographyPreamble{instructions}  
\AtEndBibliography{instructions}
```

Dans certains cas, il est utile d'exécuter des instructions selon le préambule de la bibliographie ou immédiatement avant la fin de la bibliographie, ce qui est possible à l'aide de ces deux états.

Exemple : vous voulez que la bibliographie ne soit pas justifiée, mais placée dans un flottant, à gauche. Cela se fait facilement avec :

```
\AfterBibliographyPreamble{\raggedright}
```

Vous pouvez utiliser cette déclaration à tout moment avant la bibliographie. Cependant, il est recommandé qu'elle figure dans le préambule du document ou écrivez votre propre paq.

La réalisation de cette déclaration nécessite l'utilisation d'un paq défini pour l'environnement bibliographie et compatible avec le paq correspondant (voir l'article 21.2).

## 2.24 Index

L'index est également connu sous l'appellation de registre. En ce qui concerne l'utilisation générale d'un mot-clé d'index, se référer à [DGS+12], [Lam87] et [Keh97].

Si un paq est utilisé pour fournir des auto-instructions et environnements de l'index, il atténue et même supprime éventuellement l'influence de KOMA-Script dans ce répertoire. Cela vaut, par exemple, lorsque vous utilisez l'index, mais pas lors de l'utilisation de `splitidx` (voir [Koh14b]).

```
index=réglage
```

Par défaut, `index=default` est un chapitre non numéroté sans entrée dans la table des matières. L'index est généralement la dernière partie d'un livre ou d'un autre document, et n'exige, en fait, aucune entrée dans une table des matières, mais, si cela s'avérait nécessaire, par exemple, parce qu'une procédure dans des instructions demande un index multi-unité, l'entrée peut être réalisée avec `index=totoc`. Et si l'index, contre toute pratique, doit être numéroté, vous utiliserez l'option `index=numbered`.

Voir aussi l'option `toc` avec l'index valeur ou `indexnumbered section 2.9`.

Un résumé des valeurs possibles pour le réglage de l'index peut être trouvé dans la table 2.23.

<code>\setindexpreamble{préambule}</code>
---

Analogue au préambule de la bibliographie, vous pouvez également ajouter l'index avec un préambule. Ceci trouve son application, s'il existe plus d'un index ou si différents types de référencement sont marqués, par différentes mises en évidence des numéros de pages dans l'index.

TABLE 2.23 – Valeurs possibles de l'option pour le réglage de l'index

---

`default, nottoc, plainheading`

L'index n'a pas d'entrée dans la table des matières.

---

`numbered, toctocnumbered`

L'index a une entrée numérotée dans la table des matières.

---

`toc, toc, notnumbered`

L'index a une entrée non numérotée dans la table des matières.

---

Exemple : vous avez un document dans lequel les termes sont définis et utilisés à la fois. Les numéros de page des définitions sont en **gras** et vous souhaitez signaler ce fait. Vous insérez alors dans le préambule de l'index :

```
\setindexpreamble {Les numéros de pages imprimés en
\teg{gras} sont des références à la définition du
terme particulier, contrairement aux numéros imprimés
avec le nouvel élément pertinent de la page imprimée
normale.}\Par
```

Noter que, pour la première page de l'index, le style de page donné par la macro `\indexpagestyle` est activé par défaut (voir la section 2.12). La création, le choix et la sortie de l'index sont réalisés par des paqs  $\LaTeX$  standard et des programmes add-on en sont responsables.



# Chapitre 3

## La lettre `scrlettr2`

Les lettres sont à bien des égards très différentes des articles, rapports et livres, ce qui pourrait justifier un chapitre distinct pour la classe lettre. Mais une autre raison motive un chapitre sur `scrlettr2` : cette classe, développée à partir de zéro, offre une interface différente de toutes les autres classes connues. Le type d'opération peut être rare, mais l'auteur est convaincu de proposer à l'utilisateur expérimenté de nouveaux avantages certains.

### 3.1 Les variables

Mis à part les options, commandes, environnements, compteurs et extensions, des éléments supplémentaires ont déjà été introduits au chapitre trois, dans *KOMA-Script*. Une propriété typique d'un élément est le choix d'un style de police et la possibilité d'en changer (voir rubrique 4.9).

À ce stade, nous introduisons maintenant des variables.

Les variables ont un contenu et un nom sous lequel elles sont appelées. Le contenu d'une variable est réglé dans le temps et l'espace, indépendamment de l'utilisation réelle, de manière à ce que ce contenu puisse être séparé de son exécution. La principale différence entre variable et commande : la commande déclenche habituellement une action, alors que la

variable se compose de texte brut exploité par une commande. De plus, une variable peut en outre avoir une description supplémentaire réglable.

Cette `section` est une introduction spécifique au concept de variable. Les exemples utilisés pour l'illustrer n'ont pas de signification particulière. Vous pouvez en trouver de plus détaillés dans l'explication à propos des variables prédéfinies de la classe lettre dans les sections suivantes. Ci-dessous, un aperçu des variables de `scrlttr2` dans la table 3.1.

TABLE 3.1 – Liste des variables compatibles avec `scrlttr2`

Éléments	Description
<code>addresseeimage</code>	Instruction utilisée pour imprimer le destinataire de l'option <code>addrfield=Image</code> port-payé ou de l'option <code>addrfield=backgroundimage</code> ( <a href="#">section 4.10</a> )
<code>backaddress</code>	Adresse de retour pour des enveloppes à fenêtre ( <a href="#">section 4.10</a> )
<code>backaddressseparator</code>	Séparateur dans l'adresse de retour ( <a href="#">article 4.10</a> )
<code>ccseparatot</code>	Séparateur entre titre et destinataire(s) supplémentaire(s) ( <a href="#">section 4.7</a> )
<code>customer</code>	Champ de l'occupation de ligne « numéro de client » ( <a href="#">section 4.10</a> )
<code>date</code>	Date ( <a href="#">section 4.10</a> )
<code>emailseparator</code>	Séparateur entre le nom e-mail et l'adresse e-mail ( <a href="#">section 4.10</a> )
<code>enclseparator</code>	Séparateur entre le titre et les pièces jointes ( <a href="#">section 4.7</a> )
<code>faxseparator</code>	Séparateur entre le titre de fax, et le numéro de fax ( <a href="#">section 4.10</a> )
<code>firstfoot</code>	Pied de page du papier à lettre ( <a href="#">article 4.10</a> )
<code>firsthead</code>	Entête de page du papier à lettre ( <a href="#">article 4.10</a> )
<code>fromaddress</code>	Adresse de l'expéditeur sans le nom de l'expéditeur ( <a href="#">section 4.10</a> )
<code>frombank</code>	Compte bancaire de l'expéditeur ( <a href="#">article 4.10</a> )

---

<code>fromemail</code>	Adresse e-mail de l'expéditeur (section 4.10)
<code>fromfax</code>	Numéro de fax de l'expéditeur (section 4.10)
<code>fromlogo</code>	Commandes pour insérer le logo de l'expéditeur (article 4.10)
<code>frommobilephone</code>	Numéro de téléphone portable de l'expéditeur (article 4.10)
<code>fromName</code>	Nom complet de l'expéditeur (article 4.10)
<code>fromphone</code>	Numéro de téléphone de l'expéditeur (article 4.10)
<code>fromurl</code>	URL de l'expéditeur (article 4.10)
<code>fromzipcode</code>	Code postal de l'expéditeur ou code postal de l'émetteur utilisé comme option port-payé <code>addrfield=PP</code> (section 4.10)
<code>invoice</code>	Champ des secteurs d'activité « numéro de la facture » (article 4.10)
<code>location</code>	Plus de détails sur l'expéditeur (article 4.10)
<code>mobilephoneseparator</code>	Séparateur entre le numéro de téléphone portable et l'identifiant (section 4.10)
<code>myref</code>	Référence de l'expéditeur (article 4.10)
<code>nextfoot</code>	Rubriques de bas de page utilisant des titres de style <code>headings</code> ou <code>myheadings</code> (section 4.13)
<code>nexthead</code>	Entête de page utilisant des rubriques de style <code>headings</code> ou <code>myheadings</code> (section 4.13)
<code>phoneseparator</code>	Séparateur entre le titre et le numéro de téléphone (section 4.10)
<code>place</code>	Place de l'expéditeur utilisé près de la date (article 4.10)
<code>placeseparator</code>	Séparateur entre le lieu et la date (section 4.10)
<code>PPdatamatrix</code>	Instruction affichant les données de l'option <code>addrfield=PP</code> (section 4.10)

<code>PPcode</code>	Instructions pour le code d'identification de l'expéditeur de l'option <code>addrfield=PP</code> ( <a href="#">section 4.10</a> )
<code>signature</code>	Signature et salutations à la fin de la lettre ( <a href="#">section 4.20</a> )
<code>specialmail</code>	Mode d'emploi ( <a href="#">section 4.10</a> )
<code>subject</code>	Objet de la lettre ( <a href="#">article 4.10</a> )
<code>subjectseparator</code>	Séparateur entre le titre de l'objet et l'objet ( <a href="#">article 4.10</a> )
<code>title</code>	Titre de la lettre ( <a href="#">article 4.10</a> )
<code>toaddress</code>	Adresse du destinataire, sans nom de destinataire ( <a href="#">section 4.10</a> )
<code>toname</code>	Nom complet du destinataire ( <a href="#">section 4.10</a> )
<code>yourmail</code>	Date de référence du mail au destinataire ( <a href="#">section 4.10</a> )
<code>yourref</code>	Référence du destinataire ( <a href="#">section 4.10</a> )
<code>zipcodeseparator</code>	Séparateur entre le titre du code postal et le code lui-même ( <a href="#">article 4.10</a> )

<pre>\setkomavar[désignation]{contenu} \setkomavar*{name}{désignation}</pre>
--

Le contenu du nom de variable est défini avec la déclaration `\setkomavar`. Vous pouvez en modifier la description en utilisant les arguments optionnels. En revanche, `\setkomavar*` ne peut définir la description du nom de variable. Dans les lettres, il est courant d'avoir l'expéditeur en tête. Supposons que vous ayez défini une ligne directe et que vous souhaitez en préciser le contenu. Vous écrivez :

```
\setkomavar{fromname}{Papiray}
```

l'expression par défaut pour le nom de l'expéditeur est « de ». Mais PAPIRAY veut plutôt utiliser parfois le terme « expéditeur », alors ce serait également :

```
\setkomavar*{fromname}{expéditeur}
```

Mais :

```
\setkomavar{fromname}[expéditeur]{Papiray}
```



qui définit les deux en une seule commande permet de faire d'une pierre deux coups. Soit dit en passant, le contenu de la variable peut être effacé avec un argument vide. Vous pouvez, de la même manière, supprimer la description en utilisant un argument `description` vide.

Supposons que `PAPIRAY` n'ait pas d'énoncé pour l'expéditeur. Il peut alors soit le compléter par son propre nom, ou bien le supprimer avec :

```
\setkomavar*{fromname}{}
```

Mais il peut aussi faire d'une pierre deux coups en utilisant :

```
\setkomavar{fromname}[ ]{Papiray}
```

Ainsi, le contenu de la variable et son nom sont effacés dans le même temps.

<pre>\usekomavar[commande]{nom}</pre> <pre>\usekomavar*[commande]{nom}</pre>
--

Dans certains cas, il est nécessaire que l'utilisateur accède au contenu ou à la description d'une variable d'où l'importance de ne pas laisser à la classe seule l'accès à ces fonctions. Ceci s'applique en particulier lorsque vous avez défini une variable qui n'est pas ajoutée à la ligne de champs de référence. Avec l'utilisation de la commande `\usekomavar` vous avez accès au contenu du nom de variable, considérant que la version étoilée `\usekomavar*` permet l'accès à la description ou au titre. Pour plus d'informations sur la définition de vos propres variables, voir le paragraphe 21.2.

<pre>\ifkomavar{nom}{alors la partie}{sinon else}</pre>
---

Cette commande peut être utilisée pour tester si oui ou non une variable est déjà définie. Le code sera exécuté uniquement si la variable existe déjà. Le contenu de la variable ne sera pas examiné, il peut donc être vide. Un faux code sera exécuté si la variable n'existe pas encore. Ces tests peuvent par exemple être utiles si vos propres variables sont définies dans un fichier `lco` (voir l'article 21.4 et suivants), mais utilisées uniquement dans un autre fichier `lco`.

<pre>\ifkomavareempty{nom}{alors la partie}{sinon else} \ifkomavareempty*{nom}{alors la partie}{sinon else}</pre>
---

L'utilisation de ces instructions permet de déterminer si le contenu ou la description d'une variable est vide ou non. La partie est exécutée uniquement si le contenu ou le nom élargi du nom de la variable est vide. Dans le cas contraire, la partie `else` est exécutée. La version non étoilée gère le contenu tandis que la version étoilée gère l'identifiant de variables.

## 3.2 Les pseudo-longueurs

Le processus longueur de  $\text{\LaTeX}$  est traité avec les commandes `\newlength`, `\setlength`, `\addtolength` et `\thelength`. Beaucoup de paqs utilisent ainsi des macros qui sont des ordres gérant des longueurs de stockage. `KOMA-Script` étend cette méthode à la capacité de traiter ces longueurs stockées dans des macros avec des instructions similaires aux commandes ci-dessus. Elles ne s'appliquent qu'à des longueurs réelles et prennent le nom de pseudo-longueurs.

Une liste de toutes les pseudo-longueurs `scr1ttr2` est donnée dans la table 21.1 page 574 avec leurs significations représentées graphiquement. Les dimensions utilisées correspondent à la configuration par défaut de `scr1ttr2`. Un aperçu plus détaillé des pseudo-longueurs individuelles se trouve dans les différentes sections de ce chapitre.

Normalement, les utilisateurs ne définissent pas leurs propres pseudo-longueurs. Réservée plutôt aux utilisateurs avancés, cette opération est décrite article 21.1, dans la partie expert .

Rappelez-vous que les pseudo-longueurs sont implémentées en interne comme les macros, mais que leurs noms doivent être précisés dans les instructions de leur utilisation. Leur nom ne commence pas par un contrôle<sup>1</sup>, semblable aux noms des compteurs, contrairement aux macros ou aux longueurs de  $\text{\LaTeX}$ .

---

1. Voir la note de la page xxv.

```
\useplength{name}
```

Grâce à cette commande, vous pouvez accéder à la valeur de la pseudo-longueur avec le nom donné. C'est l'une des rares commandes de l'utilisateur dans le cadre de pseudo-longueurs. Cette commande peut être utilisée, bien sûr, avec un fichier lco (voir l'article 4.21).

```
\setlengthtoplenght[facteur]{longueur}{pseudo longueur}
\addtolengthplenght[facteur]{longueur}{pseudo longueur}
```

Alors que vous pouvez simplement faire précéder un facteur de longueur par un préfixe, ce n'est pas possible avec les pseudo-longueurs. Donnons la valeur 2 pt à la longueur `\TestLength`. Puis `\TestLength` avec une valeur de 3 pt vous donnera la valeur 6 pt. Si vous utilisez à la place des pseudo-longueurs, `3 \useplength{test}` donnera 32 pt, ce qui est gênant pour attribuer une longueur réelle à la valeur d'une pseudo-longueur. Avec l'utilisation de la commande `\setlengthtoplenght` vous affecterez le multiple d'une longueur réelle à une pseudo-longueur. Ici, au lieu de précéder le facteur de la pseudo-longueur, il est donné comme nom de l'argument optionnel. Vous devez utiliser cette commande lorsque vous voulez assigner à une longueur la valeur négative d'une pseudo-longueur. Dans ce cas, vous pouvez utiliser un signe moins ou -1 comme facteur. La commande `\addtolengthplenght` fonctionne de manière similaire et ajoute le multiple d'une pseudo-longueur à la longueur pour prendre la valeur d'une pseudo-longueur.

### 3.3 Option de sélection précoce ou tardive

Tout ce qui est décrit dans la section 2.4 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris cette section vous pouvez continuer à la section 3.4.

Cette section décrit une caractéristique du programme présenté qui concerne non seulement `srlttr2` mais aussi les autres classes et paqs KOMA-Script. Pour permettre à l'utilisateur de découvrir toutes les informations sur un paq ou une classe, cette section se trouve presque à l'identique dans plusieurs chapitres. elle peut être lue dans l'un des chapitres et ignorée dans les autres.

```
\documentclass[liste d'options]{KOMA-Script-classe}  
\usepackage[liste d'options]{liste de paqs}
```

L<sup>A</sup>T<sub>E</sub>X prévoit que les utilisateurs spécifient les options de classe sous la forme d'une simple liste de mots-clés séparés par des virgules comme argument de `\documentclass`, transmises à tous les paqs qui comprennent ces options. De même, il est envisagé que les utilisateurs spécifient les options de paq sous une forme de simple liste de mots-clés comme argument optionnel de `\usepackage`. KOMA-Script étend le mécanisme des options pour les classes et quelques paqs à d'autres possibilités. Ainsi, la plupart ont une valeur supplémentaire, n'existent pas que sous la forme d'option et peuvent aussi prendre la forme `option=valeur`. Jusqu'à cette modification `\documentclass` et `\usepackage` travaillent avec KOMA-Script comme décrit dans [Tea05b] ou dans chaque introduction de L<sup>A</sup>T<sub>E</sub>X, par exemple [DGS12].

Si vous utilisez une classe KOMA-Script aucune option n'est spécifiée lors du chargement des paqs `typearea` ou `scrbase`. Les classes les chargent mais L<sup>A</sup>T<sub>E</sub>X refuse de charger à répétition un paq contenant plusieurs options. Avec `\documentclass` et `\usepackage` les paramètres longueurs, compteurs, etc. peuvent facilement casser et doivent être surveillés. L'usage de l'option longueur de L<sup>A</sup>T<sub>E</sub>X provoque un message d'erreur avant que la valeur de KOMA-Script ne soit adoptée. L'affectation de valeurs avec des longueurs ou des compteurs de L<sup>A</sup>T<sub>E</sub>X ne doit jamais précéder `\documentclass` ou `\usepackage`, mais suivre les instructions documentées `\KOMAOPTIONS` ou `\KOMAoption`.

```
\KOMAOPTIONS{liste d'options}  
\KOMAoption{option}{liste des valeurs}
```

KOMA-Script offre la possibilité de modifier la valeur d'option dans la plupart des classes et des paqs avec l'instruction `\KOMAOPTIONS`, qui permet de donner la valeur de l'option à chaque option de la liste d'options.

Certaines options ont aussi une valeur par défaut (*Default Value* en anglais). Si une valeur d'option est omise, la valeur par défaut est automatiquement appliquée.

Certaines options peuvent contenir simultanément plusieurs valeurs. Il est alors possible de leur assigner, avec `\KOMAoption` une option séquentielle d'un ensemble de valeurs. Les valeurs individuelles sont, dans la liste

de valeurs, séparées les unes des autres par des virgules. Si une valeur contient un signe égal ou une virgule, alors la valeur est mis entre accolades.

Pour la réalisation de cette possibilité de commandes KOMA-Script utilise `\FamilyOptions` et `\FamilyOption` avec la famille « KOMA », voir dans la Partie II, la section 11.3.

## 3.4 Compatibilité avec les versions antérieures

Ce qui est écrit dans la section 2.5 est applicable *mutatis mutandis*. Toutefois, cette possibilité de `scrlettr2` existe depuis la version 2.9t. Donc, si vous avez déjà lu et compris la section 2.5 vous pouvez passer ce chapitre.

Dans certains cas, des améliorations ou des corrections appliquées à la classe provoquent des changements de comportement, en particulier pendant la mise en page et ne sont pas toujours souhaitables.

<code>version=<i>wert</i></code>
<code>version=<i>first</i></code>
<code>version=<i>last</i></code>

Depuis la version 2.9t le choix est de savoir s'il est préférable de conserver, dans la mesure du possible, un fichier source ou si les ajustements à la dernière version de la classe conduisent exactement au même résultat. Il devrait être la compatibilité de version pour produire, est ainsi mis en version sur l'option. Une compatibilité de la version KOMA-Script plus ancienne peut tout d'abord être prise en charge avec la `version=first` ou la `version=2.9` ou encore la `version=2.9t` être atteint. Lorsque vous spécifiez une version inconnue en tant que valeur, un avertissement est émis et le système adopte `version=first` par précaution.

Le paramètre `version=last` charge la dernière version qui est sélectionnée et dans ce cas, néglige une compatibilité future. L'option sans valeur est `last` par défaut, aussi longtemps qu'aucune option obsolète n'est utilisée.

Si vous utilisez une option KOMA-Script2 obsolète, KOMA-Script3 la convertit automatiquement en `version=first` et dans ce cas, un avertis-

sement est émis qui explique comment vous pouvez éviter cette commutation et sélectionner une version différente de l'option de réglage, même après l'option obsolète elle-même.

La question de la compatibilité concerne principalement des problèmes de mise en page. De nouvelles options qui ne touchent pas les modifications sont disponibles, même si vous avez sélectionné une version plus ancienne en utilisant la compatibilité d'option. Cette dernière n'a pas d'effet sur les changements de transition résultant de l'élimination des erreurs manifestes lors de l'utilisation d'une version plus récente. Mais rester prudent et archiver le document utilisé par défaut dans la version `KOMA-Script`.

Les exemples lettres de ce chapitre utilisent toutes les opportunités disponibles dans la dernière version de `KOMA-Script`. Ces compatibilités doivent être réglées en conséquence pour charger la classe :

```
\documentclass[version=last]{sclttr2}
```

Noter ici la dernière version chargée avec la valeur symbolique sélectionnée, et n'oubliez pas que la version ne peut pas être modifiée après le chargement de la classe, le réglage de l'option avec `\KOMAoptions` ou `sclttr2.cmd.KOMAoption` n'est donc pas fourni.

## 3.5 Mode de conception

Ce qui est écrit dans la section 2.3 est applicable *mutatis mutandis*. Si vous l'avez déjà lue et comprise, continuez à la section 3.6.

De nombreuses classes et paqs, outre le mode normal, utilisent également un mode brouillon. Les différences entre ces deux modes sont aussi diverses que les classes et les paqs qui offrent cet avantage.

<code>draft=on-off-valeur</code>
----------------------------------

Cette option est une distinction entre documents d'un projet et les documents finaux. Les valeurs de marche-arrêt de la table 2.5 page 59 peuvent être l'une des valeurs standard utilisée. Lorsqu'elle est activée, dans le cas de lignes longues à l'excès, une petite boîte noire est émise à la fin de la ligne. Ces boîtes aident un œil non averti pour trouver les paragraphes à traiter manuellement mais en revanche, elles n'apparaissent pas

dans le mode par défaut `draft=false` et disparaissent souvent en utilisant le paq `MicroType` [SCH13].

## 3.6 Mise en page

Chaque page d'un document est séparée en plusieurs éléments différents de mise en page, par exemple, les marges, l'entête, le pied de page, une colonne marginale de note et les distances entre tous ces éléments. `KOMA-Script` juge la page dans son ensemble, taille du papier, zone d'affichage et côté visible. La gestion de la répartition de la page dans ces différentes parties est l'une des caractéristiques de base d'une classe. Dans `KOMA-Script`, ce travail est délégué à la zone de type d'un paq qui peut être utilisé en conjonction avec d'autres classes. Pour ce faire, chaque classe `KOMA-Script` utilise automatiquement le paq `typearea` qu'il n'est pas nécessaire de charger avec `\usepackage`. Voir aussi, à ce propos, la [section 2.3](#).

Certains paramètres de classes `KOMA-Script` influencent la mise en page. Ces effets sont documentés dans les réglages correspondants. Si vous voulez en savoir plus pour sélectionner la taille de la page, le format de papier, le type de zone, la séparation des pages dans les marges, l'usage mono-face ou recto-verso ou encore la composition sur une ou plusieurs colonnes, consulter la documentation du paq qui se trouve à partir du chapitre 2.

Normalement, il n'est fait aucune distinction entre les mises en page d'un seul côté et recto-verso. La plupart des lettres ne sont pas liées entre elles comme les pages d'un livre, et chaque page sera consultée isolément. C'est également le cas si les deux côtés de la feuille de papier sont utilisés pour l'impression. Dans une lettre, le réglage vertical est inhabituel, n'a en principe pas d'importance et ne joue aucun rôle. Néanmoins, en cas de besoin, vous trouverez la description de `\raggedbottom` et `\flushbottom` dans la [section 2.4](#).

## 3.7 Structure générale des documents de format lettre

La structure générale d'une lettre diffère de celle des autres documents. Alors qu'un document livre ne contient, en général, qu'un seul livre, un document lettre peut contenir plusieurs lettres. Comme le montre la figure 3.5 page 170, ce document se compose d'un préambule, du texte de la lettre séparé et de la conclusion. Le préambule comprend les paramètres qui concernent généralement toutes les lettres. La plupart d'entre eux peuvent donc être écrasés par les paramètres de lettres individuelles. Le seul paramètre qui ne peut être modifié dans une seule lettre est la version de `scrlttr2` compatible avec les versions antérieures (voir l'option dans la section 3.4).

Il est recommandé de placer les paramètres généraux, tels le chargement des paqs et le réglage des options, avant `\begin{document}` et les paramètres qui constituent le cadre de variables ou d'autres textes après. Ceci est d'autant plus important que `babel` (voir [Bra01]) est utilisé et que les variables dépendant de la langue de `scrlttr2` sont à modifier. La conclusion est habituellement `\end{document}`. Bien sûr, vous pouvez également insérer des commentaires supplémentaires à ce stade.

La figure 3.5 page 170 montre que la lettre est constituée d'une introduction, du corps et de la conclusion. Dans l'introduction sont définis les paramètres concernant la lettre actuelle. Il est vital que l'introduction se termine toujours par `\opening`, de même que la fermeture commence toujours par `\closing`. En option, les états des arguments de salutation et formule de politesse peuvent rester vides, mais les instructions doivent être utilisées et inclure un argument comme pour les arguments d'ouverture et de fermeture.

Noter que les paramètres sont modifiables entre chaque lettre individuelle et que ces modifications s'appliqueront à toutes les lettres suivantes. Pour des raisons de clarté et de maintenance de vos documents au format



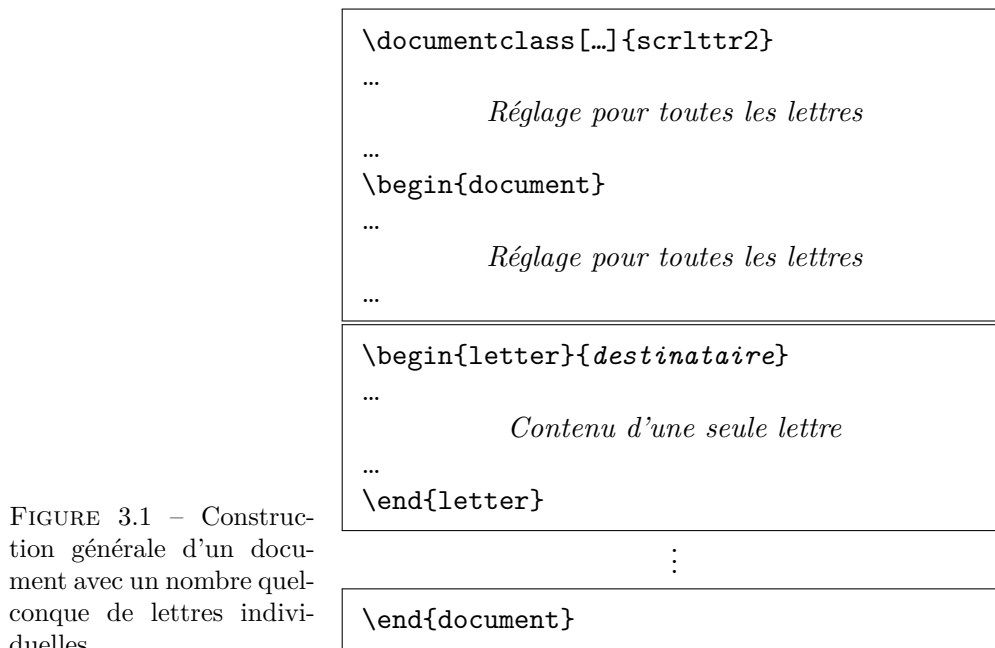


FIGURE 3.1 – Construction générale d'un document avec un nombre quelconque de lettres individuelles

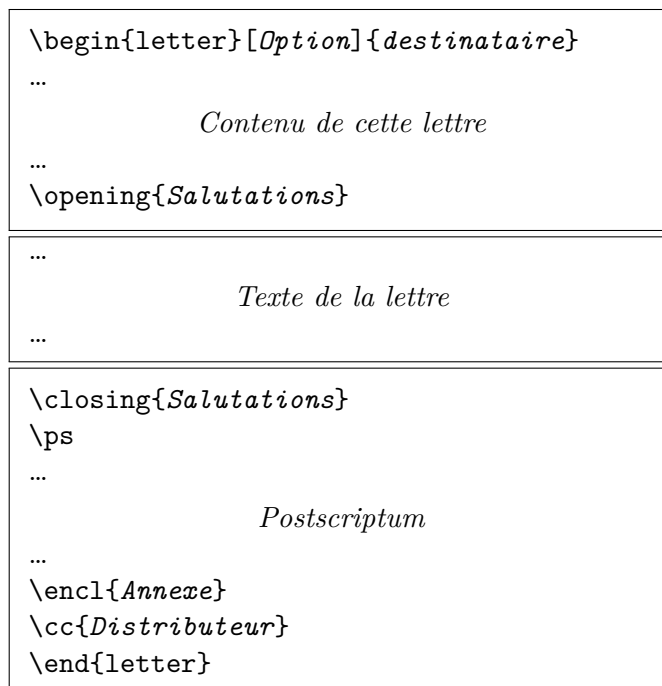


FIGURE 3.2 – Construction générale d'une seule lettre à l'intérieur d'un document de lettre

lettre, il est recommandé d'utiliser des paramètres plus généraux à validité limitée entre les lettres.

```
\begin{letter}[options]{destinataire}...\end{letter}
```

L'environnement est un des points capitales de la classe `letter`. Avec `sclttr2`, vous pouvez spécifier des options supplémentaires, des fonctions spéciales avec des arguments facultatifs, qui seront ensuite réalisées en interne à l'aide de la commande `\KOMAOptions`.

L'adresse est un argument obligatoire appliqué à l'environnement de la lettre. Une double barre oblique (backslash), inversée (antislash) ou mieux, le mot valise masculin controblique que j'adopte illico en controblic<sup>2</sup>, est utilisée comme séparateur entre les différentes parties d'adresse qui sont affichées sur une seule ligne, dans le champ `Adresse`. Néanmoins, ce double controblic ne doit pas être interprété comme un retour à la ligne. Ainsi, les éléments verticaux tels que espace ou paragraphe ne sont pas autorisés dans l'adresse car ils pourraient conduire à des résultats inattendus et des messages d'erreur, comme c'est le cas pour la classe `letter` standard.

Supposons par exemple que quelqu'un veuille écrire une lettre à PAPIRAY. Un document de lettre minimaliste pourrait ressembler à ceci :

```
\documentclass[version=last]{sclttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{Papiray \\  
chemin des vignes \\  
41250 Monvillage}
\end{letter}
\end{document}
```

Ce code n'entraîne aucune sortie imprimable et il n'y aurait pas de réponse à cette feuille de papier. La raison en sera expliquée lors de la description de la commande `\opening`.

---

2. Voir la note de la page xxv.

<pre>\AtBeginLetter{instructions} \AtEndLetter{instructions}</pre>
--

Comme mentionné dans [Tea06], il est possible d'effectuer des instructions supplémentaires en certaines occasions au cours de l'exécution de L<sup>A</sup>T<sub>E</sub>X au terme d'un document. À cette fin, les commandes utilisées les plus connues du noyau L<sup>A</sup>T<sub>E</sub>X sont `\AtBeginDocument` et `\AtEndOfClass`. On appelle ces points d'interventions des crochets, mais aussi des apostrophes.

La classe `scrlettr2` en fournit deux autres : `\AtBeginLetter` et `\AtEndLetter` permettant de déclarer le code d'instruction pour la synthèse. À l'origine, les crochets ont été créés pour le paq et la classe auteurs, de sorte qu'ils sont seulement documentés dans [Tea06] et non pas dans [Tea05b]. Cependant, des applications utiles de `\AtBeginLetter` existent avec des lettres que l'exemple suivant peut illustrer.

Supposons que vous ayez plusieurs lettres avec des questionnaires à définir dans un document. Les questions sont numérotées automatiquement dans les lettres simples en utilisant un compteur. Contrairement à la numérotation des pages, ce compteur n'est pas reconnu depuis `scrlettr2`. Il ne serait donc pas réinitialisé au début de chaque nouvelle lettre. Si chaque lettre contient dix questions, la première question portera le numéro onze dans la seconde lettre, le numéro trente-et-un dans la quatrième, etc... Une solution consiste à remettre le compteur à zéro au début de chaque nouvelle lettre :

```
\newcounter{question}
\newcommand{\question}[1]{%
  \refstepcounter{question}\par
  \noindent\begin{tabularx}{\TeXwidth}{1@{ } X}
  \theFrage:~ & # 1\
  \end{tabularx}%
}%
\AtBeginLetter{\setcounter{question}{0}}
```

De cette façon, la première question porte le numéro un, même dans la mille-et-unième lettre. Bien sûr, la définition de cet exemple nécessite le paq `tabularx` (voir [Car99c]).

<code>letter</code>
<code>\thisletter</code>
<code>\letterlastpage</code>

Dans le cas où plusieurs lettres sont dans un document, elles sont numérotées en interne par *KOMA-Script*. À cet effet, depuis la version 2.19, un compteur lettre est défini, incrémenté par chaque `\begin{lettre}` référencée.

Revenons à l'exemple de `\AtBeginLetter`. Au lieu de réinitialiser le compteur explicitement dans les `\begin{lettre}`, cela peut être fait implicitement par un compteur défini en fonction de la lettre :

```
\newcounter{question}[letter]
\newcommand{\question}[1]{ %
\refstepcounter{question}\par
\noindent\begin{tabularx}{\textwidth}{1@{X}}
\la question:~ & #1\\
\end{tabularx} %
} %
```

Ainsi, le compteur est remis automatiquement à zéro au début de chaque lettre, de sorte que la première question commence, dans chaque lettre, à nouveau avec le numéro un.

Si vous préférez une sortie à la valeur actuelle de la lettre, c'est possible d'une manière habituelle avec `\theletter`, comme mentionné ci-dessus, mais le compteur est également impacté. Ce mécanisme peut être défini au début d'une lettre avec `\label{nom d'étiquette}` pour le label et `\ref{nom d'étiquette}` pour l'étiquette, avec un point quelconque qui s'y réfère dans le document. Dans la lettre elle-même, on obtient le même résultat, même sans étiquette avec `\thisletter`.

Pour les étiquettes dans les séries de lettres, il est requis de donner un nom unique à chacune et pour ce faire, d'utiliser `\thisletter`. *KOMA-Script* avec `\thisletter` en interne définit une étiquette sur la dernière page de chaque lettre. Cela permet, avec `\letterlastpage` de connaître, à tout moment, le numéro de sortie de la dernière page de la lettre. Depuis que `\letterlastpage` travaille avec `\label` et `\pageref`, il a été constaté que, après plusieurs passages *L<sup>A</sup>T<sub>E</sub>X*, seuls deux ou trois sont valides la plupart

du temps. Vérifier que le message `Rerun` (relancer) est adapté à la sortie du terminal ou dans le fichier journal.

`\opening{salutations}`

La commande la plus importante dans `sclttr2` est, apparemment le titre de salutation de la lettre, par exemple, « Madame..., ou chère Madame » car, en fait, cette déclaration établit les marques de pliage, mais aussi la position des autres éléments de l'entête, adresse, ligne de champs de références, extension de l'expéditeur, l'empage, titre, et autres ensembles sous réserves. En bref, une lettre n'existe pas sans ouverture mais si vous souhaitez imprimer une lettre sans titre, vous devez utiliser la commande `\opening` avec un argument vide.

Exemple : nous décidons de prolonger l'ouverture par un titre, qui est alors donné par :

```
\documentclass[version=last]{sclttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{%
  Papiray \\
  chemin des vignes \\
  41250 Monvillage %
}
\Opening{Mon cher Président,}
...
```

Cela se traduira par l'entête dans la figure 3.8 :

```
\documentclass[version=last]{sclttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{%
  Papiray \\
  chemin des vignes \\
  41250 Monvillage %
}
\end{document}
```

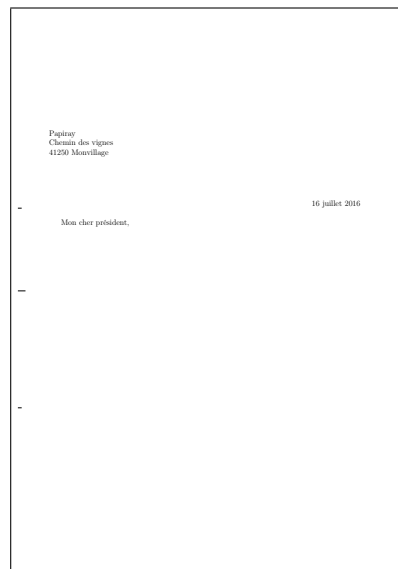


FIGURE 3.3 – Suite d'une lettre minimum avec uniquement adresse et ouverture (date et marques de pliage sont par défaut des lettres DIN)

Générées par ordinateur, les lettres omettaient le plus souvent, à l'origine, une salutation et il était à peine possible d'individualiser des lettres types. Aujourd'hui, le style personnel est d'usage pour les envois en nombre.

```
\closing{phrase de clôture}
```

Le but principal de la commande `\closing` est de créer la phrase de clôture qui peut même se composer de plusieurs lignes. Les lignes sont alors séparées par un double controblic<sup>3</sup>. Le saut de paragraphe à l'intérieur de la phrase de clôture n'est pas autorisé. En outre, cette déclaration définit également la typographie de la variable signature.

Plus de détails sur la signature et sa configuration sont donnés section 4.20.

Développons notre exemple par quelques lignes de texte et une phrase de clôture :

```
\opening{Mon cher Président,}
```

```
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos  
statuts prévoient une session au moins tous les six mois. Pour ce  
motif, je demande au conseil d'administration de réagir et d'orga-
```

---

3. Voir la note de la page xxv.

```
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\end{letter}
\end{document}
```

Ainsi, le résultat se traduira par une lettre comme celle dans la figure page 173.

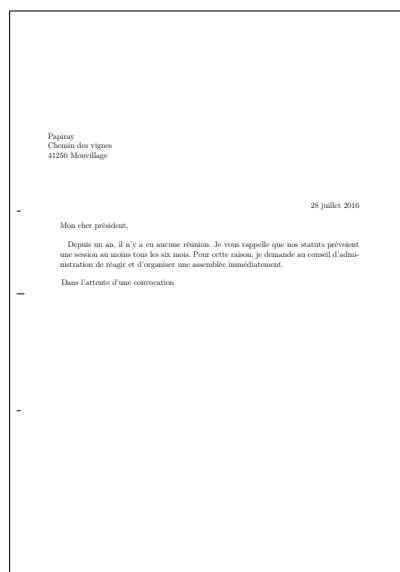


FIGURE 3.4 – Résultat d'une lettre avec adresse, ouverture, titre, texte et salutations (date et marques de pliage sont par défaut des lettres DIN)

`\ps`

L'instruction `\ps`<sup>4</sup> intègre le post-scriptum, un paragraphe commence et une distance verticale est insérée, le plus souvent en dessous de la signature. Dans la déclaration, `\ps` est suivi par du texte. L'utilisateur doit décider si le post-scriptum commence à l'abréviation `ps` qui est mise sans points. Dans `scr1ttr2`, cet acronyme n'est ni automatique ni en option.

Exemple d'une lettre prolongée par un post-scriptum, dans la figure 3.9 :

```
\documentclass[version=last]{scr1ttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{ %
```

---

4. PAPIRAY précise : veiller à utiliser `\ps` et non `\PS` qui donne le logo PS.

```
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
  Depuis un an, il n'y a eu aucune réunion. Je rappelle que nos
  statuts prévoient une session au moins tous les six mois. Pour
  cette raison, je demande au conseil d'administration de réagir et
  d'organiser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\end{letter}
\end{document}
```

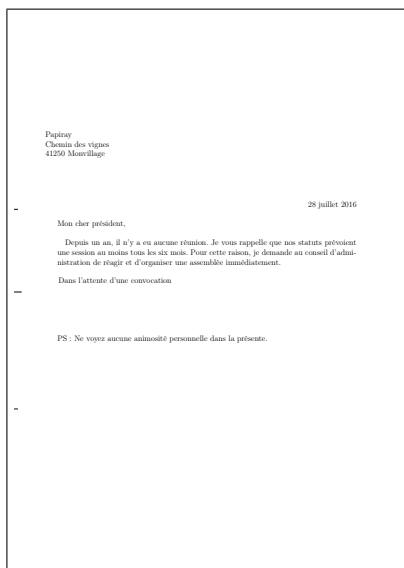


FIGURE 3.5 – Résultat d’une lettre avec adresse, ouverture, titre, texte, fermeture et post-scriptum (date et marques de pliage sont par défaut des lettres DIN)

Quand les lettres étaient écrites à la main, il était courant d’utiliser un post-scriptum, seule manière d’ajouter des informations dans la partie principale de la lettre. Bien sûr, dans des lettres écrites avec  $\LaTeX$ , vous pouvez facilement insérer des lignes supplémentaires, mais il est toujours très populaire d’utiliser un post-scriptum offrant la possibilité de souligner le plus important ou de mettre l’accent sur des choses moins vitales de la lettre.



```
\cc{liste de distribution}
\setkomavar{}[description]{contenu}
```

Avec la commande `\cc` il est possible de composer une liste de distribution. La commande prend la liste de distribution comme argument. Si le contenu de la variable `ccseparator` n'est pas vide, nom et contenu de cette variable sont insérés avant la liste de distribution qui sera reprise dans ce cas, avec un retrait approprié. Il est recommandé de définir la distribution avec `\raggedright` et de séparer les lignes par un double contrôle <sup>5</sup>.

Exemple de la lettre qui devrait, cette fois, être adressée au Président, mais aussi à tous les membres de l'association :

```
\documentclass[version=last]{sclttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{%
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Le résultat est montré dans la figure page 182. Face à la liste de distribution, l'espace <sup>6</sup> vertical est inséré automatiquement.

---

5. Voir la note de la page xxv.

6. Cet espace est masculin. L'espace féminine, qui désigne en typographie la distance entre deux lettres, est toujours sur une ligne horizontale.

### 3 La lettre `scrlettr2`

---

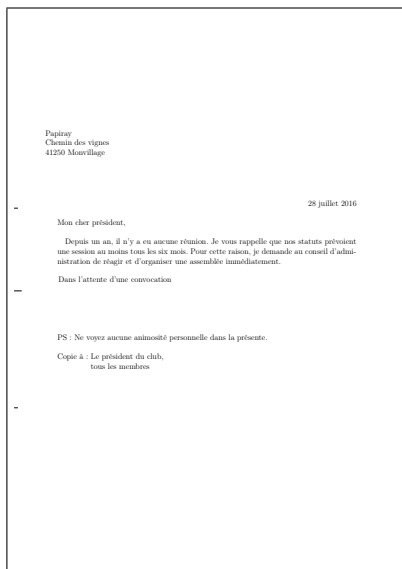


FIGURE 3.6 – Résultat d’une lettre avec adresse, ouverture, texte, fermeture, post-scriptum, et liste de distribution (date et marques de pliage sont par défaut des lettres DIN)

```
\encl{aménagements}  
\setkomavar{enclseparator}[label]{contenu}
```

Les boîtiers ont la même structure que la liste de distribution. L’ouverture seule est déterminée avec le nom et le contenu de la variable `enclseparator`.

Maintenant, la lettre est prolongée avec des paragraphes extraits des statuts qui seront ajoutés à l’ensemble. Le nom par défaut sera modifié pour se plier à un ou plusieurs systèmes.

```
\documentclass[version=last]{scrlettr2}  
\usepackage[french]{babel}  
\begin{document}  
\begin{letter}{%  
Papiray \\  
chemin des vignes \\  
41250 Monvillage %  
}  
\opening{Mon cher Président,}  
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos  
statuts prévoient une session au moins tous les six mois. Pour ce
```

motif, je demande au conseil d'administration de réagir et d'organiser une assemblée immédiatement.

```
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\pj{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Le résultat dans la figure 3.11.

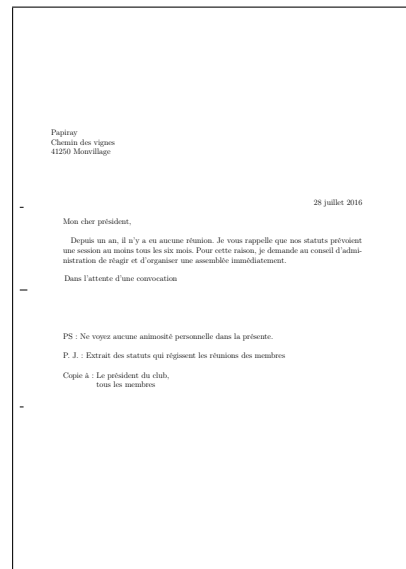


FIGURE 3.7 – Résultat d'une lettre avec adresse, ouverture, texte, fermeture, post-scriptum, pièce jointe, et liste de distribution (date et marques de pliage sont par défaut des lettres DIN)

## 3.8 Choix de la taille de police pour la lettre

La police de base et sa taille sont des éléments clés de la conception d'un document. Comme indiqué au chapitre 2, la division de la page entre la zone de texte et les marges en dépend en grande partie. La police de base est la fonte qui est utilisée pour la majeure partie du texte d'un document. Tous les paramètres de variation, forme, épaisseur, angle et taille, sont liés à cette police de base.

`fontsize=taille`

Contrairement aux classes `standard` et à la plupart des autres classes qui offrent un nombre très limité de tailles de polices, les classes `KOMA-Script` fournissent une fonction de sélection de n'importe quelle taille de police désirée pour le document principal. Dans ce contexte, une unité de mesure reconnue par  $\LaTeX$  peut être utilisée ou encore un nombre sans unité de mesure précisée, ce qui signifie pt par défaut.

Si cette option est définie dans le document, la taille de base de la police ainsi que la taille des variables liées vont changer à partir de ce point, ce qui peut être utile si une autre lettre doit être écrite dans une police générale de petite taille.

Il est à noter que la modification de taille de la police ne donne pas lieu à un calcul automatique du type de zone et des marges (voir `\recalctypearea`, section 2.4) mais que chaque recalcul se fait sur la base de la taille courante de cette police principale. L'application du changement à d'autres paqs dépend de ces paqs, mais elle entraîne parfois des messages d'erreurs typographiques qui ne peuvent pas être considérées comme fautes de `KOMA-Script`.

Cette option n'est pas destinée à être interprétée comme un substitut de `\fontsize` (voir [`Tea05a`]). En outre, vous ne devez pas l'utiliser à la place des commandes de taille de police `\tiny` jusqu'à `\Huge`. Par défaut, la taille des caractères dans *scrlltr2* est de 12pt.

Exemple : supposons que la lettre soit adressée au club des Amis des Tailles Insensées de Polices et qu'à cet effet, elle doit être imprimée en 14pt au lieu de 12pt. Ce peut être réalisé par un simple changement de la première ligne :

```
\documentclass[version=last, font size=14pt]{scrlltr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{%
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
```

```
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que
nos statuts prévoient une session au moins tous les six mois.
Pour cette raison, je demande au conseil d'administration de
réagir et d'organiser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\pj{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Sinon, la préférence peut être définie comme argument optionnel de la lettre :

```
\documentclass[version=last]{scrlettr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}[size=14pt]{ %
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que
nos statuts prévoient une session au moins tous les six mois.
Pour cette raison, je demande au conseil d'administration de
réagir et d'organiser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\pj{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Une modification tardive de taille de la police entraîne le recalcul de `typearea` et, de ce fait, les deux résultats diffèrent ci-après, dans la figure 3.12 page 191.

TABLE 3.2 – Liste alphabétique des éléments dont la police `scr1ttr2` peut être modifiée en utilisant les commandes `setkomafont` et `addtokomafont`

Éléments	Description
<code>addressee</code>	nom et l'adresse dans le champ d'adresse ( <a href="#">section 4.10</a> )
<code>backaddress</code>	adresse de retour pour une enveloppe à fenêtre ( <a href="#">section 4.10</a> )
<code>descriptionlabel</code>	étiquette de l'argument optionnel de <code>\item</code> dans un environnement détail de description ( <a href="#">section 4.16</a> )
<code>foldmark</code>	marque de pli sur le papier ; permet de changer la couleur de la ligne
<code>footnote</code>	note de pied de page et marqueur ( <a href="#">section 4.15</a> )
<code>footnotelabel</code>	marque de note, application selon l'élément note ( <a href="#">section 4.15</a> )
<code>footnotereference</code>	appel de note dans le texte ( <a href="#">section 4.15</a> )
<code>footnoterule</code>	règle horizontale au-dessus des notes à la fin de la zone de texte ( <a href="#">section 2.14</a> )
<code>fromaddress</code>	adresse de retour dans l'entête ( <a href="#">section 4.10</a> )
<code>fromname</code>	nom de l'expéditeur dans l'entête malgré <code>fromaddress</code> ( <a href="#">section 4.10</a> )
<code>fromrule</code>	ligne d'entête expéditeur destinée à des changements de couleur ( <a href="#">section 4.10</a> )
<code>labelinglabel</code>	Étiquette, c'est-à-dire argument optionnel de l'article <code>\item</code> et du séparateur dans un environnement d'étiquetage (voir la <a href="#">section 4.16</a> )
<code>labelingseparator</code>	Étiquette avec argument optionnel de <code>\item</code> dans un environnement d'étiquetage ( <a href="#">section 4.16</a> )

<code>pagefoot</code>	utilisé après l'élément <code>pageheadfoot</code> pour un pied de page défini avec la variable <code>nextfoot</code> ou pour le pied de page du paq <code>scrpage2</code> (chapitre 5)
<code>pagehead</code>	autre nom pour <code>pageheadfoot</code>
<code>pageheadfoot</code>	entête et pied de page définis dans tous les styles de page KOMA-Script (voir section 4.13)
<code>pagenumber</code>	numéro de page dans l'entête ou le pied de page (voir section 4.13)
<code>pagination</code>	autre nom pour le numéro de page
<code>placeanddate</code>	lieu et date, si une ligne de date est utilisée à la place d'une ligne de référence normale (section 4.10)
<code>refname</code>	description ou titre du champs de la ligne de référence (section 4.10)
<code>refvalue</code>	valeur du champs de la ligne de référence (section 4.10)
<code>specialmail</code>	mode d'expédition dans le champ d'adresse (section 4.10)
<code>subject</code>	objet dans l'introduction de la lettre (article 4.10)
<code>title</code>	titre dans l'introduction de la lettre (article 4.10)
<code>toaddress</code>	variation de l'élément <code>addressee</code> pour régler l'adresse du récepteur (sans nom) dans le champ d'adresse (section 4.10)
<code>toname</code>	variation de l'élément <code>addressee</code> pour le nom du destinataire dans le champ d'adresse (section 4.10)

### 3.9 Feuille de papier à lettre

L'entête est la première page et donc le fleuron d'un papier à lettre. Le secteur d'affaires utilise souvent un pré-imprimé comportant des informations, entre-autres, sur l'expéditeur et son logo. Dans KOMA-Script

ces éléments peuvent être positionnés librement. Ainsi, il est non seulement possible de reproduire directement l'entête, mais aussi de remplir les champs destinataire et adresse instantanément. Le positionnement libre est réalisé par des pseudo-longueurs (voir rubrique 4.2). Une représentation schématique de l'entête et des variables dont les noms sont imprimés en gras pour une meilleure différenciation des instructions et de leurs arguments, utilisées à cet effet, se trouvent dans la table 3.4 page 185.

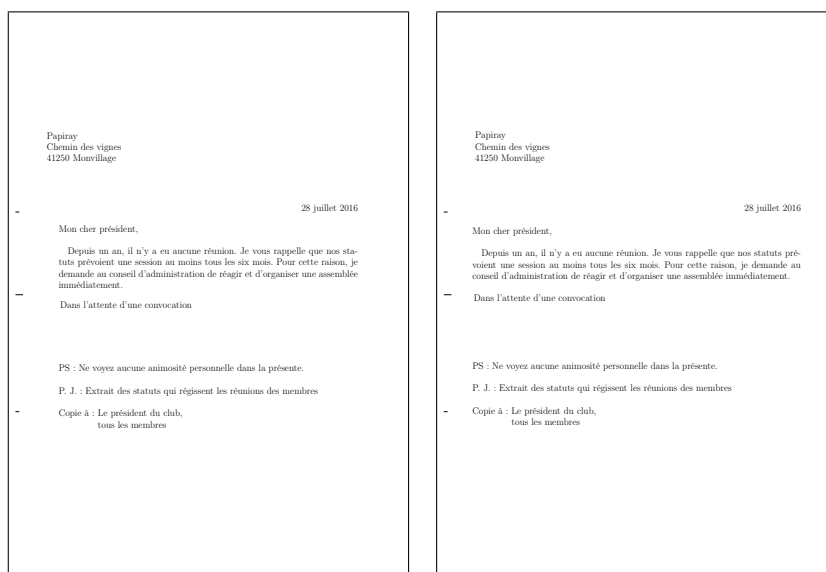


FIGURE 3.8 – Résultat d'une lettre avec adresse, titre, texte, fermeture de phrase, Post-script, installation, distribution et gros caractère de police non adaptés (date et pliages sont pris à partir des paramètres par défaut pour DIN-lettres) ; à gauche, la taille de la police a été définie comme un argument optionnel de la lettre, et à droite comme un argument optionnel de `\documentclass`

L'entête de la lettre est à distinguer, dans le langage de ce manuel, des pages suivantes qui représentent toutes les pages sauf la première.

`foldmarks=selection`

Les marques de pliage ou `foldmarks` sont de minuscules règles horizontales sur les lignes verticales gauches et de petites lignes verticales sur la marge supérieure. `KOMA-Script` offre actuellement trois marques de pliage horizontal et une marque de pliage vertical configurables. En outre, il four-



nit la marque horizontale d'un trou de perforation reconnu comme repère du milieu de page. Cette marque ne peut être déplacée verticalement.

Les options `foldmarks` permettent d'activer ou de désactiver un double, triple ou quadruple pliage vertical et une division horizontale qui ne doivent pas mener à des parties de taille égale. Les positions des quatre marques horizontales et verticales sont configurables via des pseudo-longueurs (voir section 21.1.1 et suivantes).

TABLE 3.3 – Valeurs combinables pour la configuration des marques de pliage avec l'option `foldmarks`

B	active toutes les marques horizontale supérieure bord gauche du papier
b	désactive toutes les marques horizontale supérieure bord gauche du papier
H	active toutes les marques horizontales de pliage bord gauche du papier
h	désactive toutes les marques horizontales de pliage bord gauche du papier
L	active toutes les marques verticale gauche bord supérieur du papier
l	désactive toutes les marques verticale gauche bord supérieur du papier
M	active toutes les marques horizontal milieu bord gauche du papier
m	désactive toutes les marques horizontal milieu bord gauche du papier
P	active poinçon ou marque centrale bord gauche du papier
p	désactive poinçon ou marque centrale bord gauche du papier
T	active toutes les marques horizontale inférieure bord gauche du papier
t	désactive toutes les marques horizontale inférieure bord gauche du papier
V	active toutes les marques verticales de pliage bord supérieur du papier
v	désactive toutes les marques verticales de pliage bord supérieur du papier

L'utilisateur a le choix d'appliquer les valeurs standard pour les interrupteurs simples dans le but d'activer ou désactiver toutes les marques de pliage à la fois, sur les bords gauche et supérieur du papier, ou bien il peut, en spécifiant une ou plusieurs lettres dont la liste figure dans la table 3.3, utiliser indépendamment une ou plusieurs des marques individuelles. Dans ce cas, les marques de pliage ne seront affichées que si `false`, `off` ou `no` sont généralement éteints. L'emplacement exact des marques de pliage est spécifié dans les paramètres de l'utilisateur ou des fichiers `.ico` (voir l'article

21.4) choisi pour une lettre. Les valeurs par défaut sont `true` et `TBMPL`. Vous désactiverez toutes les marques de pliage, sauf la marque de poinçonnage en utilisant par exemple :

```
\KOMAOptions{marque des plis=blmt}
```

aussi longtemps que les valeurs par défaut n'ont pas déjà été modifiées. Si des corrections ont été apportées avant, une méthode plus sûre doit être utilisée, qui change un peu notre exemple :

```
\documentclass[foldmarks=true, foldmarks=blmtP,
version=last]{scrlttr2}
\usepackage[french]{babel}
\begin{document}
\begin{letter}{%
Papiray \\\
chemin des vignes \\\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\pj{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Le résultat est montré dans la figure page 194.

L'élément `foldmark` permet de corriger la couleur de la marque de pliage à l'aide des commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9). Sa valeur par défaut est : aucun changement.

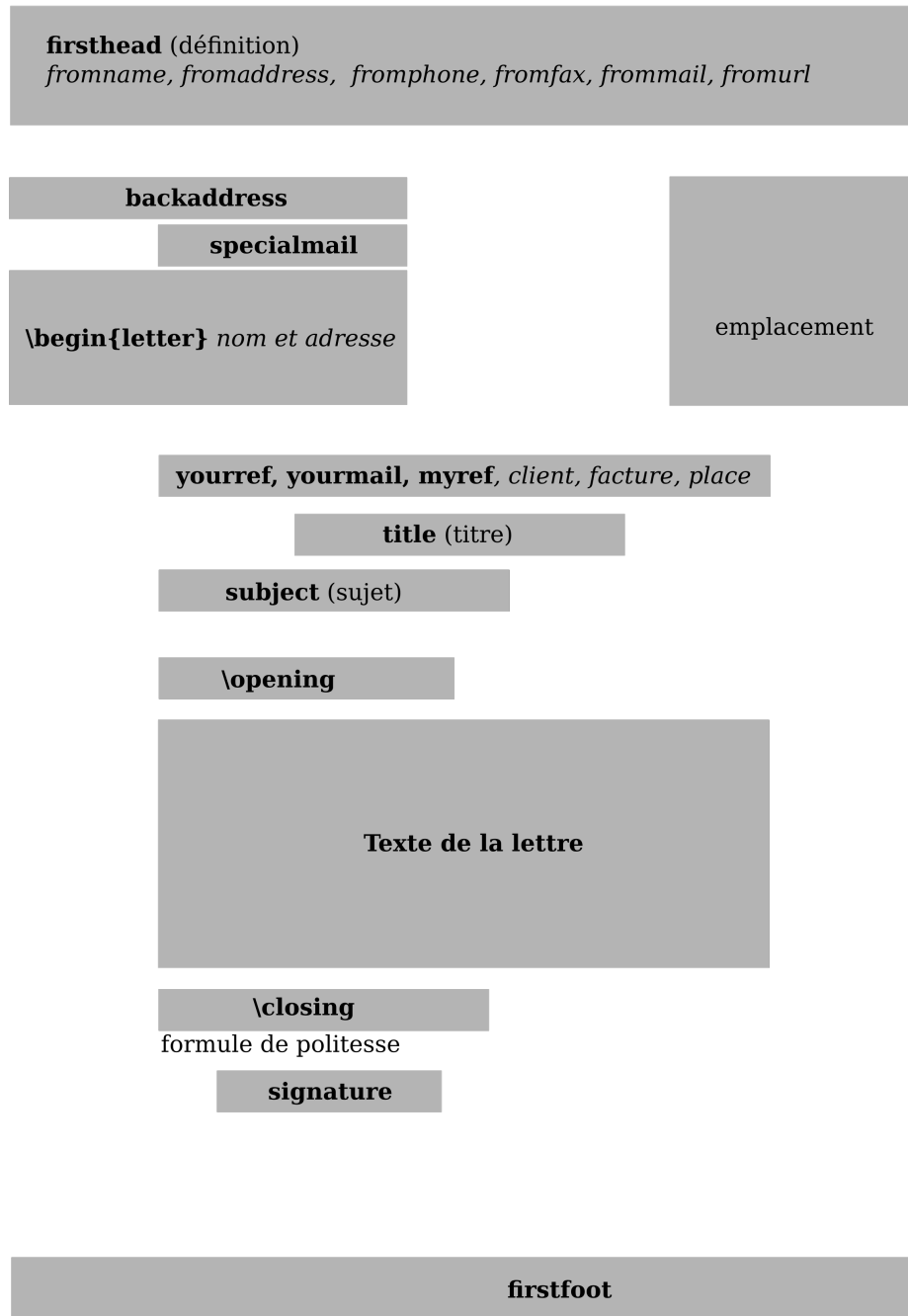


FIGURE 3.9 – Affichage schématique du papier à lettres avec les principales commandes et variables pour les éléments décrits

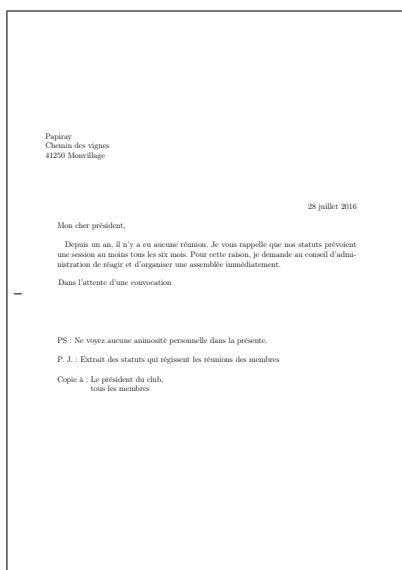


FIGURE 3.10 – Résultat d’une lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation (la date est pré-réglée pour DIN-lettres)

`enlargefirstpage=simple interrupteur`

La première page d’une lettre utilise toujours une mise en page particulière. La classe `scr1ttr2` fournit un mécanisme pour en calculer la hauteur ainsi que l’alignement vertical de l’entête et du pied de page indépendamment des pages suivantes. Par conséquent, si le pied de page de la première page atteint la zone de texte, cette dernière sera réduite en utilisant automatiquement la macro `\enlargethispage`.

De la même manière, si le pied de la première page le permet, la zone de texte devrait être automatiquement augmentée avec `\enlargethispage`. Voir aussi la description des pseudo-longueurs `firstfootvpos`. Cette option peut prendre les valeurs standard pour les simples interrupteurs, comme indiqué dans la table 1.6 page 20. Par défaut, la valeur est `false`.

`firsthead=simple interrupteur`

L’entête est généralement l’élément supérieur de la lettre. Dans `KOMA-Script`, il peut être sélectionné comme première option de la lettre, s’il doit être imprimé ou non. L’option accepte les valeurs standard `on-off` de la table 1.6 page 20 pour les touches simples. Par défaut, l’entête est activé.

**fromalign=méthode**

L'option `fromalign` détermine l'endroit où l'expéditeur est placé sur la page du début. En plus des diverses options, il est possible d'ajouter l'adresse de retour vers le poste de l'expéditeur. En outre, elle est utilisée comme interrupteur pour activer ou désactiver les extensions dans l'entête. Désactivées, les autres options n'auront pas d'effet. Les valeurs possibles pour `fromalign` sont présentées dans la table 3.4 page 185. Par défaut, l'alignement est à gauche.

**fromrule=position**

```
\setkomavar{nom de}[description]{contenu}
\setkomavar{adresse de}[description]{contenu}
```

Le nom de l'expéditeur, déterminé par la variable `fromname`, n'est pas utilisé par un entête prédéfini (voir la table 3.6 page 188).

L'entête peut être prolongé par une ligne horizontale sélectionnée à l'aide de l'option `fromrule=aftername`, après le nom. Cette barre sera toujours placée au-dessous du nom de l'expéditeur en utilisant `fromrule=afteraddress`.

La table 3.5 page 186 résume les paramètres de réglages possibles pour le trait dont la longueur, fixée par la commande pseudo-longueur `fromrulewidth`, est `false` par défaut en standard pour prolonger la règle. Sous le nom suit l'adresse de l'expéditeur déterminée par la variable `fromaddress`.

Dans l'entête la désignation des noms (voir également La table 3.6 page 188) n'est pas prédéfinie et la police utilisée pour l'expéditeur est réglable. L'écart entre le nom de l'expéditeur, le nom de l'élément et l'ensemble de la ligne peuvent être ajusté à l'aide de l'élément `fromrule`. Aucune commutation de police n'a lieu par défaut. La possibilité de commutation applicable à la ligne est principalement destinée à changer sa couleur en utilisant le gris au lieu du noir. Voir [Ker07]. Donnons maintenant le nom de l'expéditeur :

```
\documentclass[foldmarks=true, foldmarks=blmtP,
fromalign=false,
version=last]{scr1ttr2}
\usepackage[french]{babel}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
```

```
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
\begin{letter}{%
Papiray \\
chemin des vignes \\
41250 Monvillage %
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Ce n'est pas l'entête étendu mais seulement l'entête standard qui est d'abord utilisé.

Le résultat est visible, côté gauche de la figure dans la figure 3.15. Le côté droit montre presque la même lettre, mais avec `fromalign=center`, donc avec un entête élargi. Vous pouvez voir que cette variation est sans aucune règle, et dans la figure page 194, une signature apparaît maintenant sous la phrase de clôture d'abord dans le cadre du message d'accueil. Elle a été générée automatiquement à partir du nom de l'expéditeur. Plus d'informations sur la configuration de la signature peuvent être trouvées dans l'article 4.20.

Une note importante concerne l'adresse de l'expéditeur : dans cette adresse, les pièces telles que rue, code postal, ville, accessoirement état... pays... sont séparées par un double controblic<sup>7</sup>. Selon la façon dont l'adresse de l'expéditeur est utilisée, cette double barre sera interprétée différemment et n'est donc pas strictement toujours un retour à la ligne. Paragraphes, espacement vertical et autres ne sont pas autorisés dans la déclai-

---

7. Voir la note de la page xxv.

TABLE 3.4 – Valeurs possibles de l’option `fromalign` pour définir la position de l’adresse dans l’entête de `sclttr2`

<b>center, centered, middle</b>
L’expéditeur est centré dans l’entête de lettre, un logo optionnel est placé au début de l’adresse de retour, les extensions compatibles de l’entête sont activées.
<b>false, no, off</b>
Une forme standard est utilisée pour l’expéditeur, les extensions de conception de l’entête sont désactivées.
<b>left</b>
L’expéditeur est justifié, dans l’entête, à gauche et un logo est placé, en option, justifié à droite. Les extensions de l’entête sont activées.
<b>locationleft, leftlocation</b>
L’expéditeur est justifié à gauche dans l’extension de l’expéditeur, un logo en option sera placé au-dessus. L’entête est automatiquement désactivée mais peut être réactivée via l’option <code>firsthead</code> .
<b>locationright, rightlocation, location</b>
L’expéditeur est justifié à droite dans l’extension de l’expéditeur, un logo en option sera placé au-dessus. L’entête est automatiquement désactivée mais peut être réactivée via l’option <code>firsthead</code> .
<b>right</b>
L’expéditeur est justifié à droite dans l’entête, un logo en option sera placé aligné à gauche. Les extensions dans l’entête seront activées.

ration de l’adresse de l’expéditeur. Une bonne connaissance de `sclttr2` est indispensable pour utiliser intelligemment les fonctions telles que celles mentionnées ci-dessus. Un autre point à noter : il est important de définir soi-même les variables pour l’adresse de retour (voir la variable `backaddress`) et la signature (voir la variable `signature`).

La lettre a ensuite permis l’extension de l’entête en utilisant l’option `fromrule` pour obtenir une ligne sous le nom de l’expéditeur :

```
\documentclass[foldmarks=true, foldmarks=blmtP,
fromalign=center, fromrule=aftername,
```

TABLE 3.5 – Valeurs possibles de l’option `fromrule` pour la position d’une ligne horizontale dans `fromaddress` avec `scrlttr2`

<code>afteraddress, below, on, true, yes</code>
Ligne au-dessous de l’adresse de retour à l’expéditeur
<code>aftername</code>
Ligne directement en dessous du nom de l’expéditeur
<code>false, on, off</code>
pas de ligne

```

version=last]{scrlttr2}
\usepackage[french]{babel}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
\begin{letter}{%
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Dans ce
but, je demande au conseil d'administration d'organiser une assem-
blée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}

```

Le résultat est montré à gauche, et en comparaison, à droite, un exemple avec un entête standard.



### 3.9 Feuille de papier à lettre

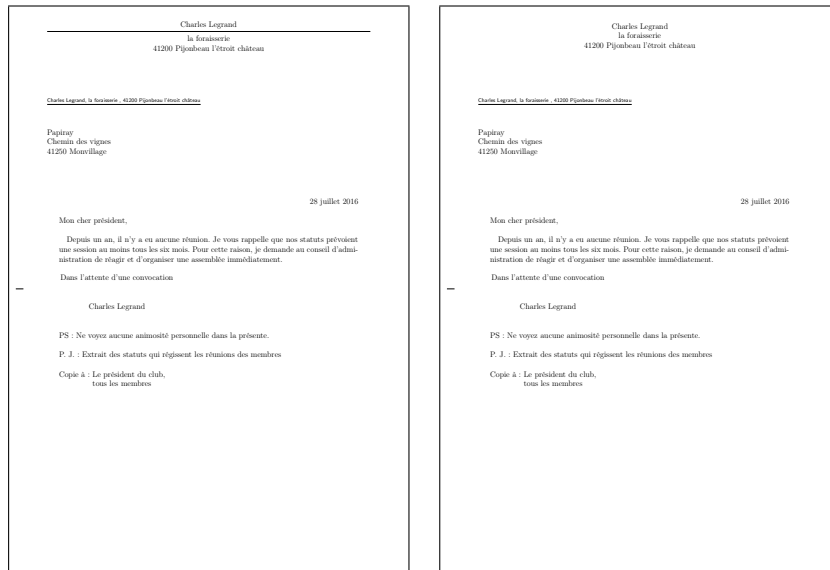


FIGURE 3.11 – Résultat d'une lettre de l'expéditeur, adresse, ligne de partage, titre, texte, phrase de fermeture, signature, post-scriptum, pièce jointe, distribution et marque de poinçon (la date provient des préférences DIN-lettres)

Avec l'aide des options `fromphone`, `frommobilephone`, `fromfax`, `fromemail` et `fromurl`, on peut déterminer si le numéro de téléphone, numéro de portable, numéro de fax, adresse e-mail ou l'URL sera l'adresse de retour. La valeur `on-off` peut être utilisée pour de simples interrupteurs dans la table 1.6 page 20 comme l'une des valeurs par défaut (qui est en réalité `false` par défaut). Les contenus sont eux-mêmes déterminés par les mêmes variables. Les paramètres pour les termes utilisés par défaut sont indiqués dans la table 3.6 page suivante, et les séparateurs utilisés et insérés entre le nom et le contenu d'une variable se trouvent dans la table 3.7 page 189.

```

symbolicnames=on-off-valeur ponctuelle
fromphone=on-off-valeur ponctuelle
frommobilephone=on-off-valeur ponctuelle
fromfax=on-off-valeur ponctuelle
fromemail=on-off-valeur ponctuelle
fromurl=on-off-valeur ponctuelle
\setkomavar{fromphone}[désignation]{contenu}
\setkomavar{frommobilephone}[désignation]{contenu}
\setkomavar{fromfax}[désignation]{contenu}
\setkomavar{fromemail}[désignation]{contenu}
\setkomavar{fromurl}[désignation]{contenu}
symbolicnames=on-off-valeur ponctuelle
\setkomavar{phoneseparator}[désignation]{contenu}
\setkomavar{mobilephoneseparator}[désignation]{contenu}
\setkomavar{faxseparator}[désignation]{contenu}
\setkomavar{emailseparator}[désignation]{contenu}
\setkomavar{urlseparator}[désignation]{contenu}

```

TABLE 3.6 – Étiquettes prédéfinies des variables pour les informations de l'expéditeur dans l'entête (nom et contenu des variables utilisées pour les délimiteurs sont indiqués dans la table 3.1 page 152)

---

fromemail	
\usekomavar*{emailseparator}	\usekomavar{emailseparator}
fromfax	
\usekomavar*{faxseparator}	\usekomavar{faxseparator}
frommobilephone	
\usekomavar*{mobilephoneseparator}	\usekomavar{mobilephoneseparator}
fromname	
\headfromname	
fromphone	
\usekomavar*{phoneseparator}	\usekomavar{phoneseparator}
fromurl	
\usekomavar*{urlseparator}	\usekomavar{urlseparator}

---

L'option `symbolicnames` permet de changer la valeur par défaut d'un seul coup. Elle peut prendre les valeurs de marche-arrêt pour simple interrupteur, comme indiqué dans la table 1.6 page 20. Son activation remplace les images de la langue et utilise les symboles du paq `marvosym` au lieu de `\mobilphonename`, `\emailname`, `\faxname` et `\phonename`. En même temps, le double point est omis dans la définition du séparateur. Pour l'url, la description et le contenu du séparateur sont négligés et resteront vides. Il est à noter que le chargement du paq `marvosym` n'est pas automatique, qu'il faut passer la première fois par `symbolicnames` et que l'option n'est activée qu'après `\begin{document}`.

TABLE 3.7 – Étiquettes et contenu du séparateur pour informations d'expéditeur pré-définies dans l'entête sans option `symbolicnumbers`

Nom	Désignation	Contenu
<code>emailseparator</code>	<code>\emailname</code>	:~
<code>faxseparator</code>	<code>\faxname</code>	:~
<code>mobilephoneseparator</code>	<code>\mobilephonename</code>	<code>\usekomavar{phoneseparator}</code>
<code>phoneseparator</code>	<code>\phonename</code>	:~
<code>urlseparator</code>	<code>\wwwname</code>	:~

Si M. Legrand a un téléphone et une adresse e-mail qu'il veut indiquer dans l'entête de sa lettre et que, de plus, il veut placer une ligne de séparation en dessous de l'entête, il utilise les options correspondantes et définit le contenu des variables nécessaires :

```
\documentclass[foldmarks=true, foldmarks=blmtP,
fromalign=false, fromrule=afteradresse,
fromphone,fromemail,
version=last]{scr1ttr2}
\usepackage[french]{babel}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
```

```
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\begin{letter}{ %
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\
tous les membres}
\end{letter}
\end{document}
```

Le résultat de cette figure est décevant, côté gauche : les options sont ignorées. C'est le cas parce que les variables et les options supplémentaires servent avec l'entête prolongé seulement. Donc, l'option `fromalign` doit être utilisée, comme le fait la bonne lettre de la figure 3.16 page 202 :

```
\documentclass[foldmarks=true, foldmarks=blmtP,
fromalign=center, fromrule=afteradress,
fromphone,fromemail,
version=last]{scrlttr2}
\usepackage[french]{babel}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\begin{letter}{ %
```

### 3.9 Feuille de papier à lettre

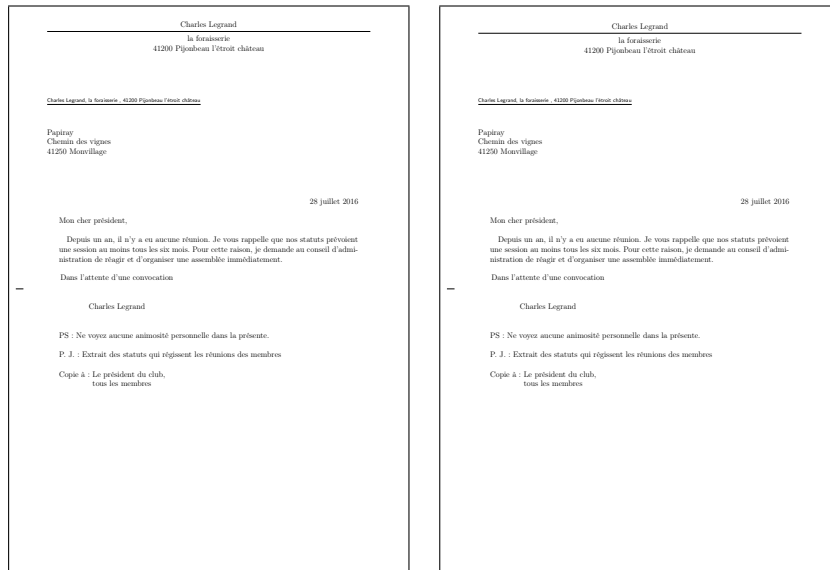


FIGURE 3.12 – Résultat d’une lettre de l’expéditeur élargi, ligne de partage, adresse, ouverture, texte, phrase de fermeture, signature, post-scriptum, pièce jointe, liste de diffusion et marque de poinçon (la date provient des préférences pour DIN-lettres)

```
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

### 3 La lettre scr1ttr2

Une comparaison de deux solutions de rechange avec l'expéditeur aligné à gauche par une position de `align=left` et l'expéditeur justifié à droite en mettant `fromalign=right` dans la figure 3.17.

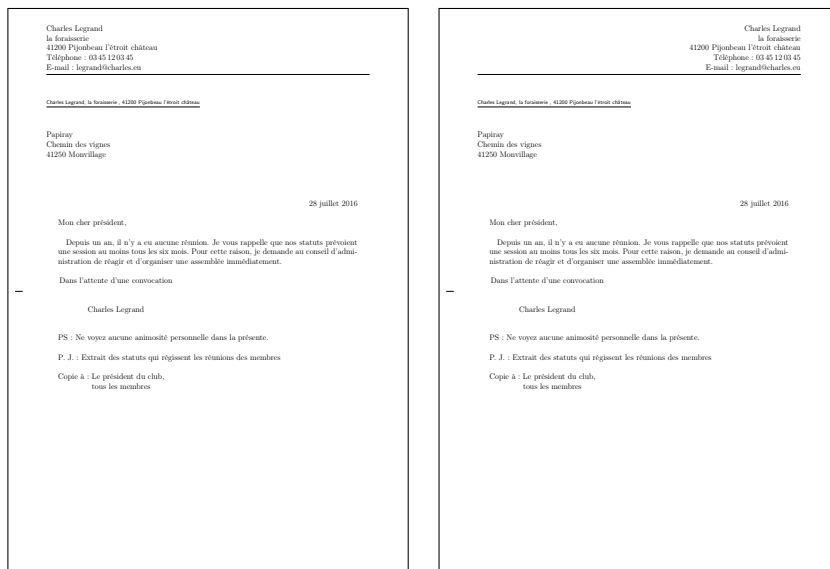


FIGURE 3.13 – Résultat d'une lettre de l'expéditeur élargi, ligne de partage, adresse, ouverture, texte, fermeture de la phrase, signature, post-scriptum, liste de diffusion et marque de poinçon (la date provient des préférences pour DIN-lettres)

```
fromlogo=valeur de marche-arrêt  
\setkomavar{fromlogo}[label]{contenu}
```

La variable `fromlogo` permet de configurer un logo placé dans l'entête avec l'une des valeurs `on-off` qui peut être utilisée (voir la table 3.1 page 152) mais le paramètre d'origine de la valeur est `false` et la description du logo est vide par défaut. **KOMA-Script** ne peut donc pas l'utiliser n'importe où sur les papiers de note prédéfinis.

Si M. Legrand considère particulièrement chic de fournir ses lettres avec un logo. Il a généré un fichier graphique qu'il veut inclure en utilisant la commande `\includegraphics`. À cet effet, il utilise le paq `graphics` (voir [Car05]) :

```
\documentclass
```

---

```

[foldmarks=true, foldmarks=blmtP,
fromrule=afteradress,
fromphone,fromemail,fromlogo,
version=last]{scrlttr2}
\usepackage[french]{babel}
\usepackage{graphics}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\setkomavar{fromlogo}{\includegraphics{legrandlogo}}
\begin{letter}{%
Papiray \\
chemin des vignes \\
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle que nos
statuts prévoient une session au moins tous les six mois. Pour ce
motif, je demande au conseil d'administration de réagir et d'orga-
niser une assemblée immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts régissant les réunions des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}

```

Le résultat est présenté dans la figure 3.14 page suivante. Les deux figures suivantes, voir la figure page suivante, montrent les résultats avec l'expéditeur centré et avec l'expéditeur en alignement en haut, à droite.

### 3 La lettre scr1tr2

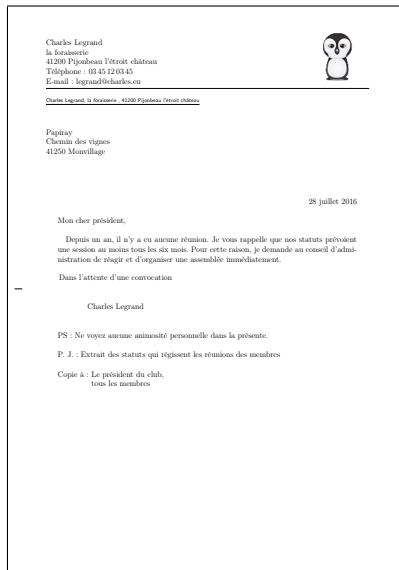


FIGURE 3.14 – Résultat d'une lettre avec l'expéditeur aligné à gauche et le logo aligné à droite)

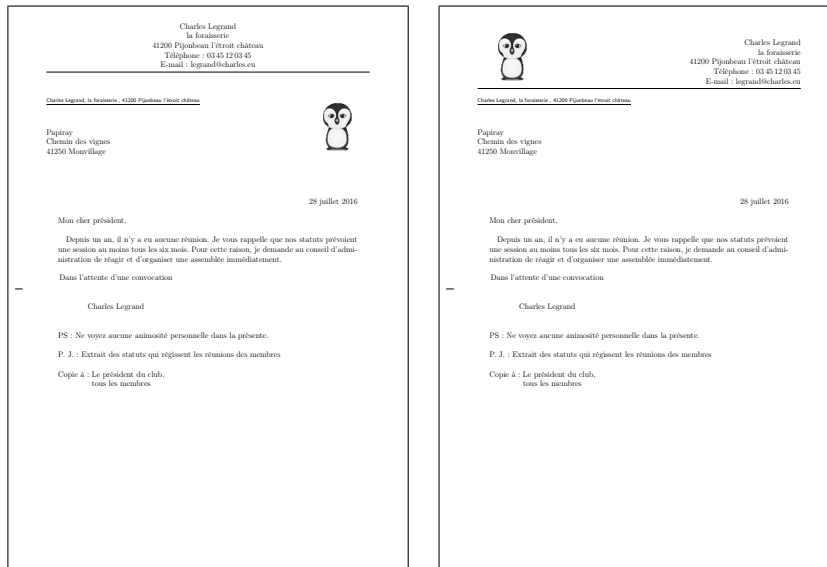


FIGURE 3.15 – Résultat avec l'expéditeur centré, logo en dessous à droite, et expéditeur aligné à droite avec logo à gauche)



`\setkomavar{firsthead}{label}{contenu}`

Dans de nombreux cas, `scr1ttr2` offre suffisamment de possibilités, avec ses options et les variables, de création d'un entête. Dans de très rares cas, vous pourriez souhaiter plus de liberté en termes de mise en page et avoir besoin de créer votre propre entête à partir de zéro. Il vous faudra alors renoncer au papier à lettres prédéfini, et vous devrez créer votre construction préférée en tant que contenu de la variable `firsthead`. Avec l'aide de `\parbox` (voir [Tea05b]), on peut concevoir plusieurs boîtes côte à côte ou l'une sous l'autre. Un utilisateur avancé sera donc en mesure de composer l'entête personnalisé dont il rêve. Bien sûr, la construction peut et doit adopter une autre variable à l'aide de `\usekomavar`.

KOMA-Script n'utilise pas la description de la variable `firsthead` qui n'existe que pour des raisons de compatibilité avec les anciennes versions de `scr1ttr2`. Obsolète, elle ne doit plus être utilisée.

L'option `Addrfield` définit si un champ d'adresse est utilisé par défaut, elle peut prendre les valeurs de la table 3.8 page 197 et son paramétrage est `valeur`. Avec cette valeur, `PP`, le nom de l'image de fond (`backgroundimage`) et l'adresse de réception sont définis par l'argument de l'environnement de lettre obligatoire (voir section 4.7) et sont en outre copiés dans les variables `toname` et `toaddress`. Les styles de polices prédéfinies peuvent être modifiés lors de l'exécution de la commande `\setkomafont` ou `\addtokomafont` (voir rubrique 4.9). Ainsi trois éléments existent dont le premier `addressee` est responsable de l'adresse et dont les éléments supplémentaires `toname` et `toaddress` ne concernent que le nom et l'adresse du destinataire et peuvent être utilisés pour définir les modifications de la configuration de l'adresse.

```
addrfield=mode
backaddress=valeur de
priority=priorité
\setkomavar{toname}[label]{contenu}
\setkomavar{toaddress}[label]{contenu}
\setkomavar{backaddress}[label]{contenu}
\setkomavar{backadresseseparator}[label]{contenu}
\setkomavar{specialmail}[label]{contenu}
\setkomavar{fromzipcode}[label]{contenu}
\setkomavar{zipcodeseparator}[label]{contenu}
\setkomavar{place}[label]{contenu}
\setkomavar{PPcode}[label]{contenu}
\setkomavar{PPdatamatrix}[label]{contenu}
\setkomavar{addresseimage}[label]{contenu}
```

L'option `addrfield` (par défaut=`true`) sert à imprimer l'adresse de retour. L'option `backaddress` définit une adresse de retour fixée pour des enveloppes à fenêtre. Cette option peut prendre les valeurs standard avec les interrupteurs simples comme indiqué dans la table 1.6 page 20. Ces valeurs ne changent pas le style de l'adresse. D'autre part, en plus de la commutation, le modèle peut être sélectionné. La valeur d'option retenue est appliquée à l'adresse de retour avec un style simple, sans soulignement, alors que par défaut, l'adresse de retour est soulignée. L'adresse de retour est fixée par le contenu de la variable `backaddress`. Par défaut, c'est une combinaison de variables de `toname` et de `toaddress` avec redéfinition du double `controblc`<sup>8</sup> pour préciser le contenu de la variable `backadresseseparator`. Le séparateur prédéfini est une virgule suivie d'une espace insécable. La description de `backaddress` variable n'est pas utilisée par `KOMA-Script`. Le style de police de l'adresse de retour est configurable via élément de `backaddress`, par défaut `\sffamily` (voir la figure page 210). Avant l'exécution de style de police, l'élément `KOMA-Script` passe en `\scriptsize`.

Par défaut, `addrfield=true` et le type d'expédition en option peuvent être émis dans le champ d'adresse entre les adresses de retour et du destinataire. Cela ne sera effectué que si la variable `specialmail` n'est pas vide

---

8. Voir la note de la page xxv.

TABLE 3.8 – Préférences pour la police des éléments de la fenêtre d’adresse

Éléments	Par défaut
<code>addressee</code>	
<code>backaddress</code>	<code>\sffamily</code>
<code>PPdata</code>	<code>\sffamily</code>
<code>PPlogo</code>	<code>\sffamily\bfseries</code>
<code>priority</code>	<code>\fontsize10pt10pt\sffamily\bfseries</code>
<code>prioritykey</code>	<code>\fontsize24.88pt24.88pt\selectfont</code>
<code>specialmail</code>	
<code>toaddress</code>	
<code>toname</code>	

et que `priority=manual` a été sélectionné, cette option est aussi la valeur par défaut. La classe `scrlettr2` n’utilise pas la représentation de la variable `specialmail`. Les pseudo-longueurs `PLengthspecialmailindent`, `specialmailrightindent` définissent la longueur de l’alignement. Le style prédéfini de police de l’élément `specialmail` peut être modifié à l’aide de la commande `\setkomafont` et de la commande `\addtokomafont` (voir rubrique 4.9).

D’autre part, en utilisant une préférence générale `priority=A` ou `priority=B` (voir la table 3.2 page 176), l’option `addrfield=true` permet d’afficher le choix pour le mode d’expédition. Utiliser `addrfield=PP` avec variables `fromzipcode` et `PPcode` extraites de la matrice de données `PPdatamatrix` fixe la priorité à Port-Payé mais n’indique pas d’adresse de retour ni le mode d’expédition. De ce fait l’élément `priority` définit le style de police de base et `prioritykey` la modification du style de police de base pour la clé de priorité, « A » ou « B ». Les styles de polices par défaut répertoriés dans la table 3.8 peuvent être modifiés à l’aide des commandes `\setkomafont` et `\addtokomafont` (voir la section 4.9).

Avec `addrfield=PP` le code postal et le contenu de la variable `fromzipcode` sont mis dans l’entête de Port-Payé. Ainsi la variable sera précédée par sa description et par le contenu de la variable `zipcodeseparator`. La descrip-

TABLE 3.9 – valeurs possibles de l’option `addrfield` pour changer le mode de destinataire dans `scrlttr2`

<code>backgroundimage,</code> <code>PPBackGroundImage,</code> <code>ppBackGroundImage</code>	<code>PPbackgroundimage,</code> <code>ppbackgroundimage,</code>	<code>PPBackgroundImage,</code> <code>ppBackgroundImage,</code>
Imprime un champ d’adresse en variable <code>adresseimage</code> avec l’entête, défini comme graphique de fond, mais sans adresse de retour ni mode d’expédition.		
<code>false, off, no</code>		
Omet le champ d’adresse.		
<code>image, Image, PPimage, PPIimage, ppimage, ppImage</code>		
Imprime <code>adresseimage</code> en variable destinataire de Port-Payé, mais ne tient pas compte des informations et des définitions pour l’adresse de retour destinataire, le mode d’expédition ou de priorité.		
<code>PP, pp, PPexplicitite, PPEexplicitite, ppexplicitite</code>		
Imprime un champ d’adresse défini par les variables <code>fromzipcode</code> , le lieu et <code>PPcode</code> , identifiées avec l’entête en priorité, et les données du tableau définies par <code>PPdatamatrix</code> mais sans adresse de retour ni mode d’expédition.		
<code>topaligned, alignedtop</code>		
Fixe une adresse avec la méthode d’expédition ou de priorité, et l’adresse de retour. L’adresse n’est pas centrée verticalement sous le mode d’expédition.		
<code>true, on, yes</code>		
Imprime un champ d’adresse avec l’adresse de retour et un mode d’expédition ou de priorité.		

tion prédéfinie est fonction du fichier `.ico` utilisé (voir l’article 4.21 et suivants). D’autre part, le contenu par défaut de la variable `zipcodeseparator` est une petite espace, suivie d’un tiret suivi par une plus petite espace (–).

En outre, avec `addrfield=PP` dans l’entête de Port-Payé, un code est défini qui identifie de manière unique l’expéditeur. Il est stocké dans la variable `PPcode`. À droite de l’adresse peut être prévue une matrice de données supplémentaires stockées dans la variable `PPdatamatrix`.

TABLE 3.10 – valeurs possibles de l’option `priority` dans le cas d’une adresse internationale

<code>false, off, no, manual</code>
Aucune priorité n’est définie
<code>B, b, economy, Economy, ECONOMY, B-ECONOMY, B-Economy, b-economy</code> Fixe la priorité B-économie internationale. Avec <code>addrfield=true</code> , se fait dans le lieu de livraison
<code>A, a, priority, Priority, PRIORITY, A-PRIORITY, A-Priority, a-priority</code> Fixe la priorité internationale. Avec <code>addrfield=true</code> , se fait à la place du service de livraison

Code postal, localité et code ont une taille de police de 8 pt par défaut. La police de l’élément `PPdata` est utilisée. Sa valeur par défaut est indiquée dans la table 3.8 page 197 et peut être modifiée à l’aide des commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9).

Avec les options `addrfield=backgroundimage` ou `addrfield=image`, l’image sera imprimée dans le champ `adresse` et `addresseeimage` sera utilisé pour stocker le contenu de la variable. `KOMA-Script` n’utilise pas la description de cette variable. L’option `addrfield=image` permet d’imprimer l’image et rien d’autre que l’image. Mais avec l’option `addrfield=backgroundimage`, le nom et l’adresse du destinataire de l’argument obligatoire de l’environnement de la lettre seront également imprimés. L’agencement de l’entête du Port-Payé est déterminé, ainsi que l’agencement de son adresse, par les pseudo longueurs `toaddrindent`, `PPheadwidth` et `PPheadheight`. L’alignement de la matrice de données est défini par `PPdatamatrixvskip`.

Noter que `KOMA-Script` ne règle pas lui-même les graphiques ou les images. Pour inclure des fichiers externes dans des variables comme `addresseeimage` ou `PPdatamatrix`, utiliser un logiciel graphique externe tel que `graphics` ou `graphicx` avec, par exemple, la commande `\includegraphics`.

**locfield=mise au point**

`scr1ttr2` place un champ avec des attributs supplémentaires de l'expéditeur à côté du champ de l'adresse. Les contenus peuvent être librement choisis, par exemple, pour le compte bancaire ou des informations similaires. Avec l'option `fromalign`, il sera aussi utilisé pour le logo de l'expéditeur. La largeur de ce champ peut être définie dans un fichier `.ico` (voir l'article 4.21). Si la largeur est réglée sur 0 dans ce fichier, alors l'option de `locfield` peut basculer entre deux pré-réglages pour la largeur du champ. Voir l'explication sur la pseudo-longueur `locwidth` à la section 21.1.4. Les valeurs possibles pour cette option sont présentés dans la table 3.4 page 185. La valeur par défaut est `narrow`.

TABLE 3.11 – Valeurs possibles de l'option `locfield` pour le réglage de la largeur du champ de l'expéditeur avec `scr1ttr2`

<code>narrow</code>
champ restreint des suppléments pour l'expéditeur
<code>wide</code>
champ large des suppléments pour l'expéditeur

**`\setkomavar{location}[désignation]{contenu}`**

Le contenu de la zone d'extension de l'expéditeur, si elle n'est pas occupée par le logo ou l'expéditeur lui-même, est déterminé par la position de la variable. Pour définir le contenu de cette variable, il est permis d'utiliser les commandes de mise en forme comme `\raggedright`. Le nom de cette variable n'est pas utilisé par `KOMA-Script`.

**`numericaldate=simple interrupteur`**

Cette option permet de basculer entre la présentation de série standard de la date dépendant de la langue et une présentation courte purement numérique. `KOMA-Script` ne fournissant pas de présentation standard, elle doit être définie par un paq tel que `french`<sup>9</sup> de `babel`, ou `isodate`. D'autre

---

9. Markus KOHM utilise `frenchb`, Pour Daniel FLIPPO, les options `frenchb` et `français` (équivalentes à `french` depuis 2004), sont conservées pour des raisons de compatibilité, mais elles ne devraient plus être en usage - voir la documentation sur le module `frenchb` de `babel`, à ce jour version v3.3b en date du 8 juillet 2017.

part, la brève présentation numérique est produite par `srlttr2`. Cette option prend des valeurs standard pour les interrupteurs simples, comme indiqué dans la table 1.6 page 20. `false` par défaut traduit une présentation standard de date.

```
\setkomavar{date}[description]{contenu}
```

La date de la présentation sélectionnée devient le contenu de la variable `date`. Le choix de l'option `numericaldate` n'influence pas la date plus longtemps, si le contenu de cette variable est modifiée par l'utilisateur. Normalement, la date sera le cadre de la ligne de référence. Si tous les autres éléments de la ligne de référence sont vides, une ligne sera donc utilisée, à la place d'une ligne de référence de date, et imprimée. Voir la description du lieu variable pour plus d'informations sur la ligne de date. Il faut noter, que vous pouvez changer l'impression automatique de la date en utilisant l'option de ligne de réf, qui sera décrite plus loin.

Si Charles Legrand tient à préciser quelques informations supplémentaires sur son adhésion à ses membres, il utilise la variable de l'expéditeur :

```
\documentclass[foldmarks=true, foldmarks=blmtP, fromphone,
fromemail, fromlogo, version=last]{srlttr2}
\usepackage[french]{babel}
\usepackage{graphics}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie
41200 Pijonbeau l'étroit château}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\setkomavar{fromlogo}{\includegraphics{legrandlogo}}
\setkomavar{location}{\raggedright
membre du club no.~40
depuis le 18.06.1970
Président 1980 - 1996}
\begin{letter}{%
Papiray
chemin des vignes
```

```
41250 Monvillage %
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle
que nos statuts prévoient une session au moins tous les
six mois. Pour ce motif, je demande au conseil
d'administration de réagir et d'organiser une assemblée
immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
}
\encl{Extrait des statuts qui régissent les réunions
des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}
```

Cela permet de définir le champ à côté de l'adresse du destinataire, comme le montre la figure 3.17.

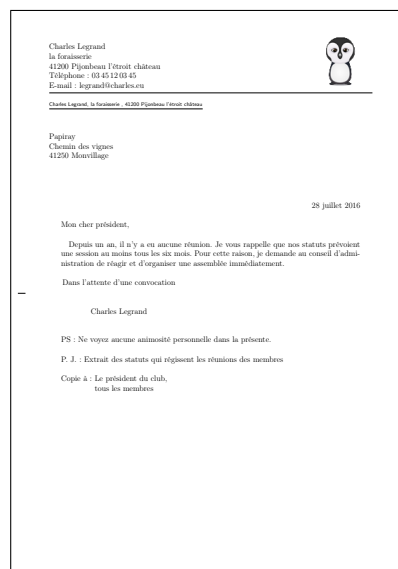


FIGURE 3.16 – Résultat d'une lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation (la date est pré-réglée pour DIN-lettres)

Dans les lettres d'affaires, on trouve souvent une ligne avec des informations telles que le code d'identification, une ligne directe de téléphone, le



TABLE 3.12 – Valeur possible de l’option `refline` pour le réglage de la largeur des champs de la ligne de référence avec `scrlttr2`

<code>dateleft</code>	La date sera placée à gauche sur la ligne de référence.
<code>dateright</code>	La date sera placée à droite à la ligne de référence.
<code>narrow</code>	La largeur de la ligne de référence sera limitée à la zone de composition.
<code>nodate</code>	La date n’est pas placée automatiquement dans la ligne de référence.
<code>wide</code>	La largeur de la ligne de référence est déterminée par l’extension de l’adresse et de l’expéditeur.

numéro de client et de facture, parfois une référence aux précédents courriers. Cette ligne sera nommée ligne de référence dans ce manuel.

`refline=selection`

Avec la classe `scrlttr2`, l’entête, le pied de page, l’adresse et les attributs de l’expéditeur peuvent s’étendre au-delà de la zone de type normal vers la gauche et vers la droite. L’option de ligne `refline=wide` définit que cela devrait s’appliquer également à la ligne des champs de références qui, normalement, contient au moins la date, mais peut contenir des données supplémentaires. Les valeurs pour cette option sont présentées dans la table 3.4 page 185, `narrow` et `dateright` sont prédéfinies par défaut.

```
\setkomavar{yourref}[description]{contenu}
\setkomavar{yourmail}[description]{contenu}
\setkomavar{myref}[description]{contenu}
\setkomavar{customer}[description]{contenu}
\setkomavar{invoice}[description]{contenu}
```

Ces cinq variables représentent les champs typiques de la ligne de référence.

Leurs significations sont données dans la table 3.6 page 188. Chaque variable possède également une description prédéfinie, indiquée dans cette même table. Des informations sur l'ajout d'autres variables à la ligne de référence peuvent être trouvées dans l'article 21.2. Le style de police, la couleur de la description et le contenu des champs de la ligne de référence peuvent être modifiés avec des éléments `refname` et `refvalue`. À cet effet les commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9) doivent être utilisées. La configuration par défaut de ces deux éléments est répertoriée dans la table 3.5 page 186.

TABLE 3.13 – descriptions prédéfinies des variables typiques de la ligne des champs de référence en utilisant des macros en fonction du langage courant

nom de variable	description	en français
<code>yourref</code>	<code>\yourref</code>	votre référence
<code>yourmail</code>	<code>\yourmailname</code>	votre courrier du
<code>myref</code>	<code>\myref</code>	notre référence
<code>customer</code>	<code>\customer</code>	n° de client
<code>invoice</code>	<code>\invoice</code>	n° de facture
<code>date</code>	<code>\date</code>	date

`\setkomavar{lieu du séparateur}[label]{contenu}`

Si toutes les variables de la ligne de référence sont vides, la ligne sera omise. Dans ce cas, le contenu de `place` et `placeseparator` seront mis, suivi par la date du jour. Le contenu prédéfini de `placeseparator` est une virgule suivie d'une espace insécable. Si le lieu de variable n'a pas de valeur de contenu, le trait d'union reste également à l'arrêt. Le contenu prédéfini de la date est `\today` et dépend de la valeur de l'option `numericaldate`. Depuis la version 3.09, lieu et date sont édités alignés à droite. Les valeurs possibles d'alignement pour cette option sont présentées dans la table 3.8 page 197. Le réglage du lieu, de la date et de la police dans la ligne de date est donné par la police de l'élément `placeanddate` de l'application `refvalue` qui est utilisée pour les lignes de référence générale. Vous pouvez changer la valeur de l'élément de police, vide par défaut, en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9).

Ainsi, monsieur Legrand pose maintenant également la variable du lieu :

```

\documentclass[foldmarks=true, foldmarks=blmtP,
fromphone,fromemail,fromlogo,
version=last]{scr1ttr2}
\usepackage[french]{babel}
\usepackage{graphics}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie
41200 Pijonbeau l'étroit château}
\begin{document}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\setkomavar{fromlogo}{\includegraphics{legrandlogo}}
\setkomavar{location}{\raggedright
membre du club no.~40
depuis le 18.06.1970
Président 1980 - 1996}
\setkomavar{date}{29. octobre 2014}
\setkomavar{place}{Pijonbeau l'étroit château}
\begin{letter}{%
Papiray
chemin des vignes
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle
que nos statuts prévoient une session au moins tous les
six mois. Pour ce motig, je demande au conseil
d'administration de réagir et d'organiser une assemblée
immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts qui régissent les réunions
des membres.}
\cc{Le président du club\\tous les membres}

```

```
\end{letter}  
\end{document}
```

Comme le montre la figure 3.3, l'endroit d'origine est affiché avant la date, suivi par la séparation automatique. La variable `date` a été explicitement définie dans l'exemple de code ci-dessus, de sorte que, lors d'une utilisation postérieure de L<sup>A</sup>T<sub>E</sub>X, c'est la date d'origine et non la date d'exécution qui sera utilisée par défaut.

TABLE 3.14 – Préférences pour la police des éléments d'occupation de ligne

Élément	Configuration par défaut
<code>refname</code>	<code>\sffamily\scriptsize</code>
<code>refvalue</code>	

```
\setkomavar{title}[label]{contenu}
```

Avec `scr1ttr2` une lettre peut porter un titre supplémentaire qui est centré et fixé avec la taille `\LARGE` directement après et sous la ligne des champs de référence. La configuration de la police prédéfinie pour le titre peut être changée avec les commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9). Les déclarations de la taille de police sont autorisées. La taille de police `\Large` ne fait pas partie de la valeur prédéfinie par défaut `\normalcolor\sffamily` mais néanmoins sera utilisée avant le style de police de l'élément.

Pour écrire un avertissement ou une mise en demeure, définissez un titre approprié :

```
\setkomavar{title}{rappel}
```

De cette manière, le destinataire reconnaîtra un rappel en tant que tel.

Bien que, comme le montre l'exemple, le contenu de la variable définisse le titre, KOMA-Script ne l'utilisera pas.

```
subject=réglage  
\setkomavar{subject}[label]{contenu}  
\setkomavar{subjectseparator}[label]{contenu}
```

Pour régler un sujet, l'aptitude d'extension de ses variables doit être définie. Sa description avant la sortie peut être configurée avec l'option

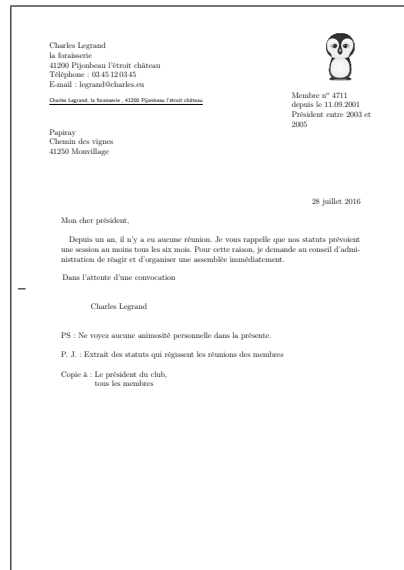


FIGURE 3.17 – Résultat d’une lettre avec logo, adresse, extension de l’expéditeur, lieu, date, sujet, titre, texte, fermeture, signature, post-scriptum, annexe, distribution et marque de pliage)

`subject=titel`, voir la table 3.7 page 189 pour la description prédéfinie. En cas d’usage de la description du contenu de `subjectseparator`, la variable sera définie entre le contenu et la description de l’objet. Le contenu de `subjectseparator` prédéfini par défaut est un deux-points suivi d’une espace.

L’option `subject=afteropening` est utilisée pour régler l’objet en dessous de la salutation avant l’ouverture de la lettre et sa mise en forme modifiée à l’aide de `subject=underlined` (souligné), ou `subject=centered` (centré), ou `subject=right` (à droite). Les valeurs disponibles sont listées dans la table 3.8 page 197. Noter que l’objet, avec `option=underlined`, doit s’inscrire sur une seule ligne. La valeur est `subject=left` (à gauche), `subject=beforeopening` (avant ouverture), et sous réserve `subject=untitled` (sans titre), par défaut.

La ligne d’objet est définie dans une police distincte. Pour la modifier, utiliser les commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9). La police prédéterminée dans `scrlttr2` est `\normalcolor\bfseries`.

Maintenant, M. Paussible établit un sujet concernant une personne souhaitant que ce sujet soit muni d’une marque lui correspondant, il définit donc une option appropriée :

TABLE 3.15 – Désignations prédéfinies de la variable pour le sujet

nom des variables prédéfinies	désignation
subject	<code>\usekomavar*{subjectseparator} %</code> <code>\usekomavar{subjectseparator}</code>
subjectseparator	<code>\subjectname</code>

```

\documentclass[foldmarks=true, foldmarks=blmtP,
  fromphone,fromemail,fromlogo,
  subject=titled,
  version=last]{scrlttr2}
\usepackage[french]{babel}
\usepackage{graphics}
\begin{document}
\setkomavar{fromname}{Alain Paussible}
\setkomavar{fromaddress}{lieu-dit Nullus
41220 Tenetour}
\setkomavar{fromphone}{03 \45 \54 \32 \10}
\setkomavar{fromemail}{alain@paussible.fr}
\setkomavar{fromlogo}{\includegraphics{alainlogo}}
\setkomavar{location}{\raggedright
membre du club no.~314\depuis le 25.11.1990
Président dans les années 1988 -1991}
\setkomavar{date}{29. octobre 2014}
\setkomavar{place}{Tenetour}
\setkomavar{subject}{Absence de réunion des membres}
\begin{letter}{%
Papiray
chemin des vignes
41250 Monvillage %
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle
que nos statuts prévoient une session au moins tous les
six mois. Pour ce motif, je demande au conseil
d'administration de réagir et d'organiser une assemblée

```

```

immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts qui régissent les réunions
des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}

```

Le résultat est montré figure 3.17 page 207.

TABLE 3.16 – Valeurs possibles pour l’option avec `scr1ttr2` sous réserve de mise en place et formatage du sujet

<code>afteropening</code>	sujet réglé après l’ouverture
<code>beforeopening</code>	sujet réglé avant l’ouverture
<code>centered</code>	sujet centré
<code>left</code>	sujet justifié à gauche
<code>right</code>	sujet justifié à droite
<code>titled</code>	description du sujet à ajouter au titre
<code>underlined</code>	ensemble soumis souligné (voir la note dans le texte)
<code>untitled</code>	ne pas ajouter à un titre

`firstfoot` = *valeur de marche-arrêt*

Cette option détermine si le pied de lettre (`letterfoot`) est réglé ou non. La coupure avec `firstfoot=false` a des effets lorsque l’option `enlargefirstpage` est simultanément utilisée. Dans ce cas, la zone de texte de la page est agrandie vers le bas. Ensuite, la distance entre la fin de la zone d’impression et le pied de la lettre devient la distance normale entre l’ensemble de la zone d’impression et le pied de page.

L’option peut prendre les valeurs standard pour les interrupteurs simples, comme indiqué dans la table 1.6 page 20. La valeur par défaut est le paramètre de composition du pied de la lettre.

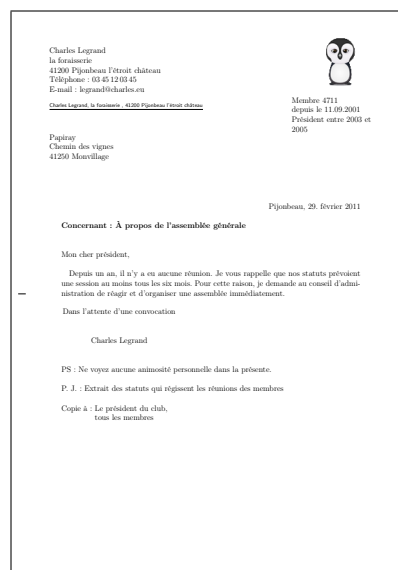


FIGURE 3.18 – Résultat d’une lettre avec logo, adresse, extension de l’expéditeur, lieu, date, marque de respect, sujet, titre, texte, fermeture, signature, post-scriptum, annexe, distribution et marque de pliage

```
\setkomavar\index{set!\setkomavar}{firstfoot}[label]{contenu}
```

Le pied de la première page est pré-réglé à vide. Il est cependant possible de faire une nouvelle construction avec les variables de `firstfoot`. KOMA-Script n’utilise pas la description de la variable. À ce sujet, voir la description de l’exemple obsolète de la commande `\firstfoot`, conservée pour des raisons de compatibilité avec les versions *scr1ttr2* antérieures à la version 3.08, mais qui ne doit pas être utilisée plus longtemps. Cette variable à ce moment prend une signification particulière : non appliquée à ce stade en interne, l’utilisateur peut s’en servir pour définir, par exemple, son compte bancaire au sein des informations complémentaires (emplacement variable) ou en pied de page.

Dans le pied de la première page, vous pouvez définir le contenu de la variable `frombank` (le compte bancaire). Le double `controblic`<sup>10</sup> doit être échangé avec une virgule dans le même temps :

```
\setkomavar{firstfoot}{ %
\parbox[b]{\linewidth}{ %
\centering\def \{\,}\usekomavar{frombank} %
```

10. Voir la note de la page xxv.



```
} %
}
```

Bien sûr, vous pouvez créer votre propre variable pour le séparateur. Je vous en laisse le soin à titre d'exercice. Si vous voulez créer un style de pied de page pour équilibrer l'entête, il peut, par exemple, être défini comme suit :

```
\setkomavar{firstfoot}{ %
  \parbox[t]{\Textwidth}{\footnotesize
  \begin{tabular}[t]{l@{}} %
    \multicolumn{1}{@{}l@{}}{actionnaires}}{titulaires}\\
  Hugo Victor \\
  Meunier Alain
  \end{tabular} %
  \hfill
  \begin{tabular}[t]{l@{}} %
    \multicolumn{1}{@{}l@{}}{gestion} \\
    Liselotte Mayer \\[1ex]
    \multicolumn{1}{@{}l@{}}{juridiction} \\
    Hinterdupfeldingen
  \end{tabular} %
  \Ifkomavareempty{frombank}{}{ %
  \hfill
  \begin{tabular}[t]{l@{}} %
    \multicolumn{1}{@{}l@{}}{ %
    \usekomavar *{frombank} :} \\
  \usekomavar{frombank}
  \end{tabular} %
} %
} %
```

L'exemple est à l'origine de Torsten KRÜGER, et avec :

```
\setkomavar{frombank}{compte bancaire12,345,678}\teg \\
Banque du Bouffon \\ %
BLZ 876 - 543 - 21
```

le compte bancaire peut être réglé en conséquence.

Dans l'exemple, le pied multiligne est fixé pour compatibilité avec la version 2.9u (voir la version dans la [section 3.4](#)), mais l'espace, en général, ne sera pas suffisant. Vous devrez alors peut-être réduire `firstfootvpos` en conséquence. La commande `\firstfoot` existe pour les mêmes raisons de compatibilité avec les versions antérieures de *scrlltr2*, mais ne doit pas être utilisée.

`\setkomavar{frombank}[label]{contenu}`

La variable `frombank` utilisée dans l'exemple précédent occupe actuellement une position particulière. Elle n'est pas encore appliquée en interne, mais peut être exploitée par l'utilisateur dans le domaine de l'extension de l'expéditeur (voir l'emplacement variable) ou définie dans le pied de l'exemple.

## 3.10 Mise en page du paragraphe

Dans l'introduction de l'article 2.10 il est expliqué pourquoi le retrait avec tiret de paragraphe est préférable à l'espacement des paragraphes. Mais les éléments de l'argumentation dépendant, par exemple, des figures, tables, listes, formules à distance et même de nouvelles pages, sont rares dans les lettres standard. En outre, les lettres ne sont généralement pas si importantes qu'un paragraphe non reconnu entraîne de graves répercussions sur la lisibilité. Les arguments sont donc plutôt faibles dans ces lettres. C'est sans doute une des raisons qui fait que l'on rencontre plus souvent des paragraphes marqués, non pas avec l'indentation de la première ligne, mais séparés entre eux par un saut vertical. Il reste donc, pour les lettres standard, deux principaux avantages de l'indentation. On peut être sûr qu'une telle lettre se distingue des autres lettres, dans la masse des lettres et qu'elle affirme, d'autre part, l'identité de l'entreprise. En dehors de ces suggestions, tout ce qui a été écrit à la [section 2.10](#) pour les autres classes *KOMA-Script* est aussi valable pour les lettres. Si vous avez déjà lu et compris la [section 2.10](#), vous pouvez passer à la [section 3.12](#) après la fin de cette [section](#).

Dans une lettre la mise en page avec l'espacement des paragraphes figure le plus souvent au lieu d'un retrait prédéterminé de paragraphe. La classe *scrlltr2* propose l'option `parskip` offrant un certain nombre de façons

d'y parvenir. Le procédé est constitué de deux parties. La première est soit `full` qui représente un espacement des paragraphes d'une ligne ou `half` pour une demi-ligne. La seconde utilise l'un des caractères « `*` », « `+` », « `-` » et peut être omise. Hors le signe de distance, la dernière ligne comporte une espace libre d'au moins `1em`, le signe `plus` réserve au moins un tiers ou l'étoile au moins un quart de la ligne en fin de ligne, mais la variante `moins` n'impose rien pour la dernière ligne d'un paragraphe.

Ce réglage peut être modifié à tout moment et dans ce cas, la déclaration `\selectfont` est exécutée implicitement. Les changements de la structure dans un paragraphe ne seront visibles que dans le paragraphe suivant.

Outre les huit combinaisons résultantes, il est toujours possible, en tant que méthode, d'utiliser les valeurs pour les interrupteurs simples de la table 1.6 page 20. Insérer une option correspond à `full` sans marque particulière pour le dégagement de la dernière ligne, avec au moins un espace carré à la fin du paragraphe. La désactivation de l'option repasse à l'indentation d'un carré. La dernière ligne d'un paragraphe peut également s'étendre à la marge de droite. Une vue d'ensemble de toutes les valeurs possibles pour la méthode est fournie dans la table 2.7 page 65.

Si un espacement de paragraphe est utilisé, la distance change avant, après et à l'intérieur de la liste des environnements, ce qui évite une plus grande séparation à partir du texte que dans du texte normal.

Dans `KOMA-Script`, `parskip=false` par défaut et il n'y a pas d'espacement entre les paragraphes, seulement une indentation de `1em`.

## 3.11 Détection des pages paires et impaires

Ce qui est décrit dans l'article 2.11 s'applique *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 2.11, vous pouvez passer à la section 3.13.

Pour les documents recto-verso, on distingue les côtés gauche et droit, un côté gauche a toujours un nombre pair et un côté droit toujours un nombre impair. Habituellement, les lettres sont fixées unilatéralement et avec une mise en page simple face ne sont jamais imprimées sur le devant

et le dos. Si à titre exceptionnel, des lettres double faces seront créées, il peut être utile de savoir si l'on se trouve sur le côté recto ou verso.

Avec *KOMA-Script*, pour trouver si un texte tombe sur une page paire ou impaire, utiliser l'instruction `ifthispageodd`. Ici, l'argument `alors` est exécuté si l'on est sur une page impaire, sinon ce sera l'argument `sinon` qui est utilisé.

Supposons par exemple que vous vouliez indiquer si un texte tombe sur une page paire ou impaire. Vous pouvez alors, par exemple, en tapant

Ceci est une page avec :

```
\ifthispageodd{un numéro de page}{}
```

avec le résultat :

```
Ceci est une page avec un numéro de page pair.
```

obtenu. Noter que dans cet exemple, l'argument `{}` (`alors`) est laissé vide.

La commande `\ifthispageodd` utilise un mécanisme similaire à `label` et `référence` qui nécessite au moins deux compilations de *LaTeX* après chaque modification pour que le résultat soit correct. Lors du premier passage, une heuristique est utilisée pour la décision.

Plus de détails sur la détection des côtés gauche et droit, impair et même sur les numéros de pages peuvent être trouvés dans le chapitre 21.1.

## 3.12 Entête et pied de page avec style prédéfini

Une des propriétés générales d'un document est le style de page. Dans *LaTeX*, cela signifie en premier lieu le contenu des entête et pied de page. Comme il est écrit dans l'article 4.10, l'entête et le pied de la feuille de lettre sont considérés comme des éléments de l'entête et par conséquent, ne dépendent pas des paramètres de style de la page. Donc, cette section décrit les styles des pages consécutives à la page d'introduction. Dans les lettres simple face, c'est le style à partir de la seconde page. Pour les lettres recto-verso c'est aussi le style de page de tous les dos.

`\letterpagestyle`

La valeur du style de la page `scrletter` est déterminée par le contenu de cette déclaration, vide par défaut. Cela signifie que la page de style correspond au reste du document, dans le sens où `scrletter2` ne gère que les lettres et qu'il est plus facile d'ajuster ce style globalement avec `\pagestyle`.

Le style ordinaire d'une page `scrletter` est différent du style brut d'une page des autres classes, et les instructions `plain.letter` sont stockées dans le paq `scrletter`. Tous les styles de page appartenant au style `plain` seront appliqués indépendamment du style de page fixé pour le reste du document. Pour de plus amples informations sur les particularités des styles de page, consulter `letterpagestyle` du paq `scrletter`.

Vous voulez utiliser `scrletter` pour que les lettres soient dans le style de page qui a été défini pour le document lui-même avec `\pagestyle`. Vous pouvez déclarer :

```
\renewcommand*{\letterpagestyle}{}
```

dans le préambule du document. Attention, l'étoile de `\renewcommand*` est importante ! Bien sûr, les déclarations `\pagestyle` et `\thispagestyle` ont la préséance sur le style de page `\begin{letter}` et `\letterpagestyle`.

```
headsepline=valeur de marche-arrêt
footsepline=simple interrupteur
```

Options pour insérer ou non une ligne de séparation en dessous de la tête ou au-dessus du pied de page, respectivement, sur les pages consécutives. L'option peut prendre les valeurs standard pour les interrupteurs simples, comme indiqué dans la table 1.6 page 20. L'activation de l'option `headsepline` commute une règle sous l'entête de la page. L'activation de l'option `footsepline` commute une règle au-dessus du pied de page. La désactivation des options désactive la ligne correspondante.

Évidemment, l'option `headsepline` n'influence pas les styles de page `empty` et `plain` (voir plus loin dans cette section). Ces styles de page ne sont pas à utiliser avec n'importe quelle entête de page.

Typographiquement, cette règle établit une connexion optique solide de l'entête ou du pied et de la zone de texte qui ne signifie pas que la distance entre l'entête et le texte ou le texte et le pied doit être augmentée, elle

devrait au contraire être considérée comme appartenant, dans le calcul du texte, au corps du texte. C'est réalisé dans **KOMA-Script** en passant l'option `headinclude` ou `footinclude` pour l'ensemble de `typearea`, si les options `headsepline` et `footsepline`, sont respectivement utilisées comme options de classe. Il en va de même pour `footsepline` et pour `footinclude`.

Au contraire, `headsepline` `footsepline` prend effet même avec la page style `plain`, car cette mise en page déplace le numéro de page dans le pied. Le paq `scrlayer-scrpage` ou `scrpage2` désuet (voir les chapitres 5 et 6) offrent plus de possibilités pour les lignes d'influence dans l'entête et le pied et peuvent aussi être combinés avec `scr1ttr2`.

`pagenumber=emplacement`

Cette option définit à quel endroit un numéro de page sera mis sur des pages consécutives, et s'il y est mis. Elle affecte le style de page `plain`, `headings` et `myheadings`. Elle a également un effet sur la page de style par défaut du paq `scrlayer-scrpage` ou la page avec un `scrpage2` désuet, si elle est définie avant de charger le paq (voir chapitre 5). Elle peut prendre des valeurs qui affectent uniquement la position horizontale, la position verticale ou influencer les deux positions. Les valeurs possibles sont énumérées dans la table 3.17 page 218. La valeur par défaut est `botcenter`.

`\pagestyle{style de page}`  
`\thispagestyle{style de cette page}`

Dans des lettres écrites avec `scr1ttr2` la distinction est faite entre quatre styles différents de page.

`empty` est le style de page, dans lequel l'entête et le pied de page de pages qui se suivent sont complètement vides. Ce style de page est également utilisé pour la première page, car l'entête et le pied de cette page sont fixés par d'autres moyens à l'aide de la macro `\opening` (voir la section 4.10).

`headings` est le style de page pour l'exécution de positions (automatiques) sur des pages consécutives. Les marques insérées à partir du nom de variable de l'objet `fromname` sont le nom de l'expéditeur et le sujet de l'objet variable (voir la section 4.10). La position à laquelle ces marques et les numéros de page sont placés, dépend de l'option `pagenumber` dé-

crite précédemment et du contenu des variables `nexthead` et `nextfoot`. L'auteur peut manuellement modifier ces marques après la commande `\opening`. À cette fin, les commandes `\markboth` et `\markright` sont disponibles comme d'habitude, et le paq `scrlayer-scrpage` ou `scrpage2` permet aussi l'utilisation de `\markleft` (voir section 5.1.2).

`myheadings` est le style manuel de page du titre des rubriques sur les pages qui se suivent. Ceci est très similaire à un titre, mais ici les marques doivent être réglées par l'auteur en utilisant les commandes `\markboth` et `\markright`. L'utilisation du paq `scrpage2` permet également d'utiliser `\markleft`.

`plain` est le style de page avec seulement les numéros de page dans l'entête (ou le pied) de pages consécutives. Le placement de ces numéros de pages est influencé par l'option décrite précédemment `pagenumber`.

Les styles de page sont également influencés par le choix `headsepline` et `footsepline` décrits précédemment. Le style commençant par la page en cours est changé à l'aide de `\pagestyle`. En retour, les changements `\thispagestyle` n'affectent que le style de la page en cours. La classe lettre elle-même utilise `\thispagestyle{empty}` dans `\opening` de la première page de la lettre.

Pour changer le style de police des entête et pied de page, vous devez utiliser l'interface utilisateur décrite dans la section 2.6. Pour l'entête et le pied de page le même élément est utilisé, que vous pouvez nommer soit `pageheadfoot` ou `pagehead`. C'est un élément `pagefoot` supplémentaire pour le pied de page. Il sera utilisé après `pageheadfoot` à la page `foot` qui a été définie soit avec la variable `nextfoot` ou le paq `scrpage2` (voir chapitre 5). L'élément pour le numéro de page dans l'entête ou le pied de page est appelé le numéro de page. Les réglages par défaut sont listés dans dans la table 3.2 page 176. Merci de consulter aussi l'exemple de la section 2.12.

```
\markboth{marque gauche}{marque droite}
\markright{marque droite}
```

Les possibilités qu'offrent les variables et options dans `scrlttr2` devraient suffire dans la plupart des cas pour créer entête et pied de page des pages consécutives à la première page de la lettre.

D'autant que vous pouvez également changer avec `\markboth` et `\markright` les déclarations de l'expéditeur que `scr1ttr2` utilise pour créer l'entête. Si vous utilisez le paq `scr1ayer-scrpage` ou `scrpage2` cela s'applique également au style de page `scrheadings`. La commande `\markleft` est en outre disponible.

```
\setkomavar{nexthead}[description]{contenu}
\setkomavar{nextfoot}[description]{contenu}
```

Parfois, on veut plus de liberté pour personnaliser l'entête ou le pied de page suivant. Dans ce cas, les options prédéfinies déclarées par l'option numéro de page sélectionnée ci-dessus peuvent être omises et l'entête et le pied des pages suivantes dans le style de page `headings` ou `myheadings` sont conçus en toute liberté, construction préférée avec le contenu de la variable `nexthead` où selon le cas `nextfoot`. Dans le contenu de `nexthead` et `nextfoot` vous pouvez avoir plusieurs boîtes côte à côte ou l'une sous l'autre en utilisant, par exemple, la commande `\parbox` (voir [Tea05b]).

Un utilisateur avancé ne devrait pas avoir de problème pour concevoir ses propres entête et pied de page. Dans le contenu vous pouvez bien sûr accéder également à d'autres variables en utilisant `\usekomavar`. `KOMA-Script` n'utilise pas la désignation de l'une ou l'autre de ces variables.

Les commandes obsolètes `\nexthead` et `\nextfoot` existent pour des raisons de compatibilité avec les versions antérieures de `scr1ttr2` mais ne doivent plus être utilisées.

TABLE 3.17 – Valeurs possibles de l'option `pagenumber` pour positionner la pagination à la page style des rubriques `myheadings` et `plain`

Éléments
<code>bot, foot</code>
Numéro de page dans le pied sans changer la position horizontale
<code>botcenter, botcentered, botmiddle, footcenter, footcentered, footmiddle</code>
Numéro de page centré à l'intérieur du pied,
<code>botleft, footleft</code>
Numéro de page dans le pied gauche
<code>botright, footright</code>
Numéro de page dans le pied droit



---

 Éléments (suite)
 

---

 center, centered, middle
 

---

 Numéro de page centré, sans modification de la position verticale
 

---

 false, no, off
 

---

 Pas de numéro de page
 

---

 head, top
 

---

 Numéro de page dans l'entête sans changer la position horizontale
 

---

 headcenter, headcentered, headmiddle, topcenter, topcentered
 

---

 topmiddle
 

---

 Numéro de page centré à l'intérieur de l'entête
 

---

 headleft, topleft
 

---

 Nombre de pages à gauche dans la tête
 

---

 headright, topright
 

---

 Numéro de page à droite dans la tête
 

---

 left
 

---

 Numéro de page à gauche sans modifier la position verticale
 

---

 right
 

---

 Numéro de page à droite sans changer la position verticale
 

---

### 3.13 Pages intercalaires

Ce qui est écrit dans la [section 2.13](#) est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la [section 2.13](#), vous pouvez continuer à la [section 3.14](#).

Des pages blanches sont des pages qui restent intentionnellement vide lorsque l'ont écrit un document, définies par défaut pour le style de page en cours de validité dans L<sup>A</sup>T<sub>E</sub>X. KOMA-Script fournit ici diverses extensions.

En général, on trouve des pages blanches principalement dans les livres car les chapitres commencent toujours sur un côté droit et, dans le cas où le chapitre précédent se termine également sur un côté droit, une page vide est insérée du côté gauche. Il existe aussi les pages vierges dans la mise en page recto-verso. Une page blanche est plutôt inhabituelle pour des lettres.

Cela tient moins à ce que les lettres vraiment recto-verso soient assez rares, plutôt qu'à l'absence de lien naturel avec les livres, mais `sclttr2` soutient, au cas où, les paramètres de pages blanches des lettres recto-verso. Les études développées trouvent difficilement une utilisation dans des lettres dont vous ne trouverez pas d'exemple ici. Si nécessaire, consulter la [section 3.13](#).

<code>cleardoublepage=style de page</code> <code>cleardoublepage=courant</code>
--

Cette option permet de déterminer le style de page vide qui est insérée lorsque il est requis par les instructions `\cleardoublepage`, `\cleardoubleoddpaper` ou `\cleardoubleevenpage` pour parvenir à la page désirée. Les styles qui sont déjà définis dans la page style (voir la section 4.13 et le chapitre 5) sont tous utilisables. En outre, `cleardoublepage=current` (courant) est actuellement possible. Ce cas qui correspond à la valeur par défaut de la version KOMA-Script 2.98c, crée la page blanche avec le style de page en cours lors de l'insertion.

Depuis la version 3.00, les pages blanches sont par défaut en conformité avec les conventions typographiques créées avec le style de page `empty` si vous n'avez pas défini une compatibilité avec les versions antérieures de KOMA-Script (voir la version de l'option, section 4.4).

Dans L<sup>A</sup>T<sub>E</sub>X-Kern existe une commande `\clearpage` qui garantit que tous les environnements flottants qui ne sont pas encore édités le seront et qui commence une nouvelle page. En outre, la commande `\cleardoublepage` qui fonctionne comme `\clearpage`, mais uniquement dans la mise en page recto-verso (voir option `twoside` dans la section 2.6), lance une nouvelle page de droite, avec une page blanche à gauche affichée dans le style de page, si nécessaire.

Avec KOMA-Script `\cleardoubleoddstandardpage` fonctionne exactement de la manière décrite pour les classes `standard`.

La commande `\cleardoubleoddplainpage` modifie le style `plain` de la page de gauche vide afin de supprimer l'entête.

La déclaration, `\cleardoubleoddemptypage` est utilisée pour supprimer à la fois l'entête et le pied de la page vide qui est maintenant complètement vide. Vous pouvez spécifier votre propre style de page pour la page blanche comme argument de `\cleardoubleoddpaperusingstyle`. Dans ce cas, tous les styles de page précédemment définis (voir aussi le chapitre 5) sont utilisés.

```

clearpage
cleardoublepage
cleardoublepageusingstyle{style de page}
cleardoubleemptypage
cleardoubleplainpage
cleardoublestandardpage
cleardoubleoddpge
cleardoubleoddpgeusingstyle{style de page}
cleardoubleoddemtpage
cleardoubleoddpainpage
cleardoubleoddpstandardpage
cleardoubleevenpage
cleardoubleevenpageusingstyle{style de page}
cleardoubleevenemtpage
cleardoubleevenplainpage
cleardoubleevenstandardpage

```

Le fonctionnement de la commande `\cleardoublestandardpage` attend un argument, ainsi que `\cleardoubleemptypage` et `\cleardoubleplainpage`.

La commande `\cleardoublepageusingstyle` adaptée à `sclrttr2` fonctionne de la manière que la commande standard `\cleardoublepage` correspondante, comme expliqué précédemment.

Les autres instructions sont définies pour l'exhaustivité. Plus d'informations sur celles-ci peuvent être trouvées [section 2.13](#).

Les commandes `\cleardoubleoddpge` et `\cleardoublepage` conduisent à la prochaine partie gauche ou à droite. Ici, le style de page est choisi en fonction de l'option `\cleardoublepage` et une page vierge est éventuellement insérée.

## 3.14 Notes de pied de page

Ce qui est écrit dans la [section 2.14](#) est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la [section 2.14](#), vous pouvez continuer à la [section 2.15](#).

Les instructions pour la mise en bas de page peuvent être trouvées dans chaque introduction `LATEX`, par exemple, `[DGS]`. `KOMA-Script` propose ce qui suit, mais permet aussi la possibilité de changer la forme des notes.

Que les notes soient autorisées pour les lettres dépend de la nature de la lettre et de sa mise en page. Par exemple, la collision optique avec la base de l'entête ou une confusion avec la liste des distributeurs ou encore des éléments similaires typiques de lettres sont à éviter, ce qui est de la responsabilité de l'utilisateur.

Bien que les notes de pied de page soient rarement utilisées dans les lettres, quelques exemples ont été incorporés à cette section. Pour en savoir davantage, consulter les notes, chapitre 3.14.

`footnotes=réglage`

Les notes de bas de page sont identifiées dans le texte avec de petits chiffres en exposant par défaut. Si un passage se compose de plusieurs notes consécutives, il peut donner l'impression qu'il ne s'agit pas de notes distinctes, mais d'une seule note avec un procédé supérieur. Avec `footnotes=multiple` les notes se succèdent directement, au lieu d'être bordées de délimiteurs. Le séparateur défini dans `multfootsep` est pré-réglé par une virgule. L'ensemble du mécanisme est compatible avec la mise en œuvre de `footmisc` version 5.3d (voir [Fai11]). Elle affecte les deux marqueurs de bas de page qui ont été fixés avec `\footnote`, ainsi que ceux émis directement avec `\footnotemark`.

Il est possible à tout moment, avec `\KOMAOPTIONS` ou `\KOMAOPTION`, de revenir aux notes par défaut `footnotes=nomultiple`. Pour des problèmes avec d'autres paqs qui influent sur les notes de cette option, l'option ne devraient pas être utilisée et ses paramètres non activés dans le document. Un résumé des valeurs possibles pour le réglage des notes est fourni dans la table 2.11 page 77.

```
\footnote[numéro]{texte}
\footnotemark[numéro]
\footnotetext[numéro]{texte}
\multiplefootnoteseparator
\multfootsep
```

Avec `KOMA-Script`, les notes se gèrent exactement comme dans les classes `standard` avec la commande `footnote` ou en utilisant la paire `\footnotemark` et `\footnotetext`. Tout comme avec les classes `standard`, il est possible que, dans une note, un saut de page se produise, généralement lorsque la marque de bas de page correspondante est placée si bas que la mise en page n'a pas d'autre choix. Par contraste avec les classes `standard`, `KOMA-Script` offre dans une succession et avec un séparateur, la possibilité de renvoi automatique directement à la page suivante. Voir les options `footnotes` déjà documentées.

Ce séparateur peut être mis en place manuellement à l'aide de la commande `\multiplefootnoteseparator`. Toutefois, cette déclaration ne devrait pas être

redéfinie par l'utilisateur car, en plus du séparateur et de sa mise en forme, elle permet le choix de la taille des caractères et contient `superscript`. Le séparateur lui-même est enregistré dans `\multifootsep`. Ce qui est défini par :

```
\newcommand*{\multifootsep}{,}
```

peut être redéfini.

Des exemples et des informations supplémentaires peuvent être trouvées dans la section 2.14.

```
\footref{référence}
```

Parfois, la même note de pied de page devrait être référencées plusieurs fois dans le texte d'un document. Une solution consisterait en l'utilisation de `\footnotemark` pour spécifier le numéro souhaité. L'inconvénient majeur de cette solution est que vous devez connaître le numéro qui peut changer à tout moment. Pour résoudre ce problème, KOMA-Script utilise `\label` incluant dans la note correspondante, une déclaration et ensuite, avec `\footref`, toutes les marques `footnote`, pour cette note, dans le texte. Étant donné que les notes de bas de page peuvent être définies en utilisant `\label`, les refontes influant cette numérotation nécessitent éventuellement une compilation double pour que les marques en soient correctes.

Voir la section 2.14 pour un exemple d'utilisation de `\footref`.

```
\deffootnote[largeur de marque]{indentation}{sous-indentation}
{définition de la marque}
\deffootnotemark{définition de la marque}
\thefootnotemark
```

Les classes KOMA-Script créent les notes d'une façon quelque peu différentes des classes `standard`. Le numéro de référence dans le texte identifie ainsi la note, comme cela se fait avec les classes `standard`, par un petit numéro en exposant. La même mise en forme est reproduite dans la note elle-même qui est justifiée à droite dans un champ de la largeur du texte du document. La première ligne de la note suit directement le numéro de marquage.

Les lignes suivantes sont émises et rédigées à l'identique. Si le paramètre en option de largeur n'est pas spécifié, la valeur par défaut est utilisée. Si la note se compose de plusieurs points, la première ligne d'un paragraphe sera en outre fournie avec une entrée indentée.

La figure page 163 illustre les différents paramètres. Le réglage par défaut dans les classes KOMA-Script correspond à la définition suivante :

```
\deffootnote[1em]{1.5em}{1em}{%  
\textsuperscript{\thefootnotemark}%  
}
```

Ici, à l'aide de `\textsuperscript` agit à la fois sur l'exposant et la taille de police plus petite. La commande `\thefootnotemark` retourne l'identifiant de la note actuelle sans aucun formatage. Merci de ne pas abuser de cet élément à d'autres fins, par exemple pour l'utilisation de justification dans les notes (voir aussi `\raggedfootnote`).

Les éléments d'application des polices personnalisées se trouvent également dans la note en plus de l'étiquetage. En utilisant les commandes `\setkomafont` et `\addtokomafont` (voir [section 3.9](#)), la police de l'étiquette peut être réglée différemment. Voir aussi la valeur par défaut de commutation de chaque police dans la [table 3.1](#) page 152.

La marque de référence dans le texte est définie séparément de la marque avant la note. Cela se fait avec la commande `\deffootnotemark`. Le réglage par défaut est :

```
\deffootnotemark{\textsuperscript{\thefootnotemark}}
```

Ainsi on trouve les paramètres de l'élément d'application `footnotereference` (voir la [table 4.2](#) page 266). Les marquages dans le texte et dans la note elle-même sont identiques. La police peut toutefois être modifiée avec les instructions `\setkomafont` et `\addtokomafont` (voir [section 4.9](#)).

Des exemples peuvent être trouvés dans la [section 2.14](#).

```
\setfootnoterule[hauteur]{longueur}
```

Habituellement, un séparateur est placé entre la zone de texte et les notes et en général ne se prolonge pas sur toute la largeur. La longueur exacte et la hauteur (ou l'épaisseur) de la ligne seront déterminées avec cette déclaration. Elles sont elles-mêmes évaluées en fonction de `\normalsize` lors de la définition de la ligne. Le paramètre optionnel d'épaisseur peut être complètement éliminé et ne sera pas modifié. Si les arguments hauteur ou longueur sont vides, la taille correspondante ne change pas. Un avertissement est émis dans le cas d'utilisation de valeurs non acceptables.

La couleur de la ligne de l'élément `footnoterule` est fixée avec les commandes `\setkomafont` et `\addtokomafont` (voir [section 4.9](#)). Aucun changement de police ou de couleur par défaut et pour modifier la couleur, le paq `xcolor` doit être chargé.

## 3.15 Listes

Ce qui est écrit dans l'article 2.18 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris l'article 2.18, vous pouvez continuer à la section 3.17.

$\LaTeX$  et les classes **standard** offrent différents environnements pour les listes. Ils sont inclus bien sûr dans **KOMA-Script**, avec parfois de légères modifications ou extensions. Fondamentalement, les listes - même de différents types - peuvent être imbriquées à une profondeur de quatre listes. Une imbrication plus profonde n'aurait, en typographie, aucun sens parce que, à proprement parler, plus de trois niveaux peut déjà être considéré comme excessif et je recommande en pareil cas de diviser une grande liste en petites listes.

Puisque les listes sont des éléments standard de  $\LaTeX$ , les exemples ont été omis dans cette section. Vous pouvez en trouver dans la section 2.18 ou dans toute présentation de  $\LaTeX$ .

```
\begin{itemize}
\item...
:
\end{itemize}
\labelitemi
\labelitemii
\labelitemiii
\labelitemiv
```

La forme la plus simple de liste est le **point-clé** (Stichpunkt) ou la liste **itemize**. Dans les classes **KOMA-Script** et suivant le niveau, « • », « - », « \* » et « · » sont utilisés pour marquer un élément de liste. La définition du caractère pour chaque niveau est stockée dans les macros `\labelitemi`, `\labelitemii`, `\labelitemiii` et `\labelitemiv`. Vous pouvez facilement la redéfinir avec `\renewcommand`. Chaque **point-clé** sera présenté avec `\item`.

La liste numérotée est également fréquente et fournie par le noyau  $\LaTeX$ . La numérotation, selon le niveau, est dans les différents caractères : chiffres arabes, lettres minuscules, les petits chiffres romains et les lettres majuscules. Le type de numérotation est défini par des macros de `\theenumi` jusqu'à `\theenumiv`. Le format de la sortie est déterminée par les macros `\labelenumi` à `\labelenumiv`.

```
\begin{enumerate}
\item...
:
\end{enumerate}
\theenumi
\theenumii
\theenumiii
\theenumiv
\labelenumi
\labelenumii
\labelenumiii
\labelenumiv
```

Dans ce cas, une parenthèse suit la valeur du second niveau, qui est sortie en minuscules, les valeurs de tous les autres niveaux sont suivies d'un point et chaque article sera de nouveau introduit avec `\item`.

```
\begin{description}
\item[liste à puce] ...
:
\end{description}
```

Une autre forme de liste est une liste de mots-clés qui sert principalement à la description de plusieurs articles qui reçoivent un paramètre optionnel dans `\item`. La police utilisée peut être modifiée pour mettre en évidence le mot-clé dans les classes KOMA-Script, en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir la section 3.9) pour l'élément `descriptionlabel` (voir la table 3.2 page 176).

Par défaut `\sffamily` et `\bfseries` sont utilisés.

```
\begin{description} [délimiter] {mot-clé le plus long}
\item [mot-clé]...
:
\end{description}
```

Une autre forme de liste disponible uniquement avec les classes KOMA-Script est l'environnement `labeling`. Contrairement au précédent contexte présenté, un modèle déterminant l'indentation de tous les éléments peut être spécifié dans l'étiquetage et un séparateur réglé entre l'article et le texte de description. La police pour mettre en évidence le mot-clé est paramétrée en utilisant les



commandes `\setkomafont` et `\addtokomafont` (voir section 4.9, à la page 180) pour modifier l'élément `labelinglabel` (voir le tableau 4.2, page 181) et pour placer un espace entre la police et l'élément séparateur `labelingseparator` (voir également le tableau 4.2, page 181). La pensée de la zone était à l'origine pour les structures comme « condition, déclaration, preuve » ou « question, recherche, solution », que l'on trouve fréquemment dans les notes de cours. En fin de compte, l'environnement s'est développé dans des applications très différentes et les exemples de ce manuel ont été définis avec l'environnement `labeling`.

```
\begin{verse}...\end{verse}
```

Le paq `verse` n'est généralement pas perçu comme un environnement de liste parce qu'il n'utilise pas `\item`. Au lieu de cela, la procédure travaille les sauts de ligne fixes dans un environnement `flushleft`. Il est cependant, en interne, dans les deux classes `standard` et c'est un environnement de liste avec `KOMA-Script`. L'environnement `verse` est utilisé pour les poèmes. Les lignes sont espacées à leur gauche et leur droite. Les vers simples se terminent donc par un saut de ligne fixe, définis chacun comme un paragraphe et séparés par une ligne blanche. Souvent, `\medskip` ou `\bigskip` sont utilisés à la place. Si vous voulez éviter qu'un saut de page se produise à la fin d'un verset, utiliser `\\*` au lieu de `\\`.

```
\begin{quote}...\end{quote}
\begin{quotation}...\end{quotation}
```

Ces deux environnements appartiennent également à la liste et se trouvent à la fois dans les classes `standard` et avec `KOMA-Script`. Ils utilisent un texte justifié, étiré à droite et à gauche. Habituellement, ils sont fréquemment utilisés pour déposer plus de citations. La différence entre les deux réside dans la manière dont les paragraphes sont interrompus. Alors que, avec `quote`, les paragraphes sont mis en évidence par l'espace vertical, c'est l'indentation horizontale de la première ligne d'un paragraphe qui est privilégiée avec `quotation`, cela vaut également pour le premier paragraphe de l'environnement. Pour empêcher cette indentation, vous devez faire précéder la commande par `\noindent`.

```
\begin{addmargin}[indentation gauche]{indentation}...
\end{addmargin}
\begin{addmargin*}[indentation gauche]{indentation}...
\end{addmargin*}
```

Comme avec `quote` et `quotation`, `addmargin` modifie la marge. Cependant, à la différence des deux premiers environnements, l'utilisateur peut choisir avec

`addmargin` la valeur de la modification. En outre, l'environnement ne change ni l'indentation de paragraphe, ni leur espacement. Aucun espacement vertical supplémentaire ne sera inséré après l'environnement.

Si seul l'argument obligatoire de l'indentation est donné, son contenu est isolé à la droite et la gauche de ce montant. L'argument optionnel d'indentation gauche donné provoque un retrait à gauche sur la gauche, malgré la collecte de la valeur ajoutée à la marge.

La version étoilée `addmargin*` ne diffère de la version sans étoile que dans le jeu `recto-verso`, la différence se produit quand l'argument interne optionnel d'indentation est utilisé. Ici, la valeur de l'empreinte interne est ajoutée à la partie de bord intérieur de la page, à la droite du bord gauche de la page et à la gauche du bord droit. L'entrée est alors valable pour chaque bord respectif.

Dans les deux variantes de l'environnement, les valeurs négatives pour tous les paramètres sont autorisées. Ainsi, on peut obtenir que le milieu soit en saillie dans la marge.

Si une page est un côté gauche ou droit ne peut pas être déterminé de manière fiable lors de la première exécution de  $\text{\LaTeX}$ . Voir à ce propos les explications des instructions `\ifthispageodd` (section 3.12) et `\ifthispagewasodd` (section 21.1).

Dans l'interaction des environnements tels que les listes de paragraphes, les profanes se posent souvent les mêmes questions. Par conséquent, une explication supplémentaire de l'option dans la section 21.1 est également consacrée à ce sujet. Dans la partie expert de cette section, le problème de plusieurs pages dans l'environnement `\addmargin*` est également traité.

## 3.16 Mathématiques

Il n'y a pas d'environnement mathématiques appliqué à `KOMA-Script`, mais les caractéristiques de mathématiques du noyau  $\text{\LaTeX}$  sont prises en charge. Les mathématiques plus régulières avec des équations ou des formules numérotées sont très rares, quasi inexistantes en lettres. Pour cette raison, `scrlettr2` ne supporte pas activement les équations numérotées et les options `leqno` et `fleqn`, décrites pour `scrbook`, `scrreprt` et `scrartcl` à la section 2.19, ne sont pas disponibles dans `scrlettr2`.

Vous ne trouverez pas, ici, de description des environnements mathématiques du noyau  $\text{\LaTeX}$ . Si vous souhaitez utiliser `displaymath`, `equation` et `eqnarray`,

vous devez lire une courte introduction  $\LaTeX$  comme [DGS12]. Mais si vous voulez plus de mathématiques très simples, l'utilisation du paq `amsmath` est recommandée (voir [Ame02]).

## 3.17 Environnements flottants de tableaux et figures

Les environnements flottants pour tableaux ou figures sont très rares dans les lettres. Par conséquent `sclrttr2` ne les fournit pas. Si pourtant ceux-ci sont nécessaires, ce qui indique souvent un usage abusif de la classe lettre, vous pouvez essayer de les définir avec l'aide d'un paq comme `tocbasic` (voir chapitre 14). Néanmoins, tableaux, figures ou graphiques sans environnement flottant ni titres sont bien sûr également possibles avec la classe `sclrttr2`.

## 3.18 Notes de marge

Ce qui est écrit dans la section 2.21 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 2.21, vous pouvez continuer au chapitre 4.20.

Outre la zone principale de texte qui remplit l'espace d'impression, la colonne dite marginale existe à l'intérieur des documents dans laquelle ce type de note peut être réglé. Cependant, les notes de marge sont plutôt rares dans les lettres et seront donc utilisées avec parcimonie.

<pre>\marginpar[<i>note de marge</i>]{<i>note de marge</i>} \marginline{<i>note de marge</i>}</pre>
---

Normalement  $\LaTeX$  fournit la commande `\marginpar` pour gérer les notes de marges. La note est placée dans le bord extérieur. Dans une page unilatérale, c'est la marge droite qui est utilisée. Même si un côté différent peut être spécifié avec l'option `\marginpar` et si la note est dans la marge gauche, les notes de marge sont toujours sorties dans l'ensemble du bloc. L'expérience montre que, au lieu de l'ensemble de blocs, il est souvent préférable de justifier les notes à gauche ou à droite. Pour ce faire, `KOMA-Script` propose la commande `\marginline`. Un exemple détaillé peut être trouvé dans la section 2.21.

Pour les experts, les problèmes rencontrés avec l'utilisation de `\marginpar` sont décrits et documentés dans la section 21.1. Ceux-ci appliquent également

à `\marginline`. En outre, un paq qui permet de réaliser sa propre pagination et d'annoter les colonnes est présenté au chapitre 19.

## 3.19 Salutations finales

Les salutations réglées par `\closing` ont déjà été expliquées dans la section 4.7. Une annotation est souvent placée en dessous de la signature de la lettre. La signature ou une inscription écrite elle-même à la main est placée entre cette annotation et la phrase de clôture.

```
\setkomavar{signature}[label]{contenu}
```

La variable signature peut contenir une annotation ou un type d'explication prédéfini avec `\usekomavar{fromname}`. Elle peut être constituée de plusieurs lignes. Les lignes doivent ensuite être séparées par un double controblic `\\`<sup>11</sup>. Les sauts de paragraphe ne sont pas autorisés dans l'annotation.

```
\raggedsignature
```

La phrase de clôture ainsi que la signature sont composées dans une boîte dont la largeur est déterminée par la ligne la plus longue de la phrase de fermeture ou du contenu de signature. Placer cette case est déterminé par les pseudo-longueurs `sigindent` et `sigbeforevskip`. La commande `\raggedsignature` détermine l'alignement dans la boîte. Avec le fichier `.ico`, la déclaration est fixée par `\centering` (tout sauf KOMAold) ou par `\raggedright` (KOMAold). Par exemple, pour obtenir à l'intérieur de la boîte un alignement justifié à gauche ou à droite, la commande peut être redéfinie comme la commande `\raggedsection` (voir l'exemple dans la section 2.16).

Exemple : maintenant, M. Legrand veut vraiment encore se grandir. À cet effet, il utilise la signature pour montrer encore une fois, qu'il était auparavant président. En raison de cela, il change le contenu de la variable signature. De plus, il veut que la signature affleure, alignée à gauche et donc redéfinit aussi `\raggedsignature` :

```
\documentclass [foldmarks=true, foldmarks=blmtP,  
fromphone, fromemail, fromlogo,  
subject=titled,  
version=last]{scr1ttr2}  
\usepackage [french]{babel}
```

---

11. Voir la note de la page xxv.

---

```

\usepackage{graphics}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{signature}{Charles Legrand
(ancien président)}
\renewcommand *{\raggedsignature}{\raggedright}
\setkomavar{fromaddress}{la foraisserie
41200 Pijonbeau l'étroit château}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\setkomavar{fromlogo}{\includegraphics{legrandlogo}}
\setkomavar{location}{\raggedright
membre du club no.~40
depuis le 18.06.1970
Président 1980 - 1996}
\setkomavar{date}{29 Février 2011}
\setkomavar{place}{Public-Village}
\setkomavar{objet}{Absence d'assemblée générale}
\begin{letter}{%
Papiray
chemin des vignes
41250 Monvillage%
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle
que nos statuts prévoient une session au moins tous les
six mois. Pour ce motif, je demande au conseil
d'administration de réagir et d'organiser une assemblée
immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\encl{Extrait des statuts qui régissent les réunions
des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}

```

Pour le résultat, voir la figure 3.19 page suivante.

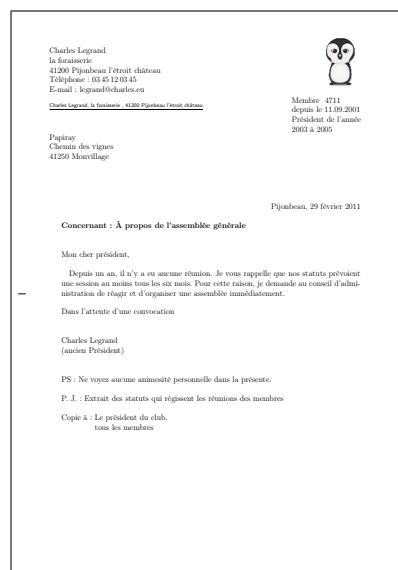


FIGURE 3.19 – Résultat d’une lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation (la date est pré-réglée pour DIN-lettres)

## 3.20 Fichiers en option de la classe `Lettre`

Normalement, vous ne devriez pas redéfinir les sélections comme informations de l’expéditeur à chaque écriture d’une lettre. Au lieu de cela, vous réutiliserez un ensemble de paramètres pour certaines occasions. Ce seront les mêmes pour l’entête et le pied de page utilisés sur la première page, ce qui rend raisonnable de mémoriser ces paramètres dans un fichier séparé. Le paq `scrlettr2` propose ici les fichiers `.ico`. Le suffixe `.ico` est une abréviation pour l’option de classe lettre.

Dans les fichiers `.ico` toutes les déclarations peuvent être utilisées, ce qui pourrait également être utilisé au point dans le document où le fichier `.ico` est chargé. En outre, des instructions internes disponibles pour les auteurs de paq, pourraient être utilisées. Dans `scrlettr2` ce sont en particulier les commandes `\@newplength`, `\@setlength` et `\@addtoplength` (voir la section 21.1).

KOMA-Script contient déjà certains fichiers `.ico`, les fichiers `DIN.ico`, `NF.ico`, `SNleft.ico`, `SN.ico`, `UScommercial9`, `UScommercial9DW` et `DINmTeXt.ico` qui permettent à `scrlettr2` de s’appliquer à des normes différentes. Ils sont adaptés comme modèles pour vos propres jeux de paramètres. Le fichier `KOMAold.ico` sert à améliorer la compatibilité de `scrlettr` avec l’ancienne classe de lettre. Comme il s’agit de recours à des instructions qui ne sont pas approuvées pour les auteurs de paq, vous ne devriez pas les utiliser comme modèles pour vos

propres fichiers `.ico`. Une liste de tous les fichiers `.ico` prédéfinis peut être trouvée dans la table 3.10 page 199.

Si vous avez créé un jeu de paramètres pour une lettre standard qui n'est pas encore prise en charge par `KOMA-Script`, vous êtes invité à l'envoyer à l'adresse du support de `KOMA-Script`. Merci de ne pas oublier d'inclure l'autorisation pour la distribution sous licence `KOMA-Script` (voir le fichier `lpp1-de.txt` du paq `KOMA-Script`). Si vous avez les mesures utiles pour une lettre standard non prise en charge, mais que vous ne désirez pas écrire vous-même un fichier `lco` correspondant, vous pouvez également contacter l'auteur, Markus KOHM. Des exemples de certains fichiers `.ico` partiellement très complexes peuvent être trouvés notamment dans [PDK] et dans [Koh03].

```
\LoadLetterOption{nom}
\LoadLetterOptions{liste de noms}
```

Habituellement, les fichiers `.ico` sont chargés directement via `\documentclass`. Entrer le nom du fichier sans le suffixe `.ico`. Le fichier `.ico` sera chargé juste après le fichier de classe. Il est toutefois recommandé de le charger ensuite, ce qui peut être fait avec les commandes `\LoadLetterOption` ou `\LoadLetterOptions`. Cela est possible à une date ultérieure même depuis un autre fichier `.ico`. Dans ce cas, le nom du fichier `.ico` est également transmis sans suffixe.

Malgré l'attente de l'argument `name` de `\LoadLetterOption` du fichier `.ico`, `\LoadLetterOptions` fournit une liste de noms séparés par des virgules. Les documents liés au nom du fichier `.ico` sont chargés dans l'ordre d'entrée dans la liste.

M. Legrand a également écrit un document contenant plusieurs lettres dont la majorité doivent se conformer à la norme DIN. Alors il commence par :

```
\documentclass{srltr2}
```

Cependant, dans une lettre, on doit utiliser la variante `DINmTeXt`, avec un champ d'adresse placé plus vers le haut, ce qui entraîne une première page de texte plus appropriée. Le pliage sera modifié de sorte que le champ d'adresse corresponde toujours avec la fenêtre dans une enveloppe à la norme DIN C6/5.

Vous pouvez atteindre cet objectif comme suit :

```
\begin{letter}{%
Papiray
chemin des vignes
41250 Monvillage %
}
```

```
\LoadLetterOption{DINmTeXt}
\opening{Bonjour,}
```

Comme la construction de la première page commence réellement avec `\opening`, il suffit de charger le fichier `lco` avant, mais le chargement ne doit pas être fait avant `\begin{letter}`. Les modifications apportées par le chargement du fichier `lco` sont locales, dans la lettre correspondante. Si un fichier `.ico` est chargé par `\documentclass`, il ne peut pas avoir le même nom qu'une option.

Depuis que M. Legrand écrit régulièrement des lettres en employant les mêmes paramètres, il trouve assez ennuyeux d'avoir, encore et encore, cette information à copier dans chaque nouvelle lettre. Ainsi écrit-il un fichier `lco` qui va faciliter son travail, à son grand soulagement :

```
\ProvidesFile{ich.lco}[2014 /06/11 lco
(Charles Legrand)]
\KOMAOptions{foldmarks = true, foldmarks = blmtP,
subject = intitulé}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{signature}{ Charles Legrand \\
(ancien Président)}
\renewcommand{\raggedsignature}{\raggedright}
\setkomavar{fromaddress}{la foraisserie \\
41200 Pijonbeau l'étroit château}
\setkomavar{fromphone}{03 \45 \12 \03 \45}
\setkomavar{fromemail}{legrand@charles.eu}
\setkomavar{fromlogo}{%
\includegraphics{legrandlogo}}
\setkomavar{location}{\raggedright
membre du club no.~40\\
depuis le 18.06.1970\\
Président 1980 - 1996}
\setkomavar{place}{Public-Village}
\setkomavar{frombank}{Banque du Bouffon \Chamfond}
```

Ainsi, sa lettre du dernier exemple diminue considérablement :

```
\documentclass[version=last, ich]{scr1ttr2}
\usepackage[french]{babel}
\usepackage{ graphics}
\begin{document}
\setkomavar{date}{29 Février 2011}
```



```

\setkomavar{subject}{réunion manquante}
\begin{letter}{%
  Papiray \\
  chemin des vignes \\
  41250 Monvillage%
}
\opening{Mon cher Président,}
Depuis un an, il n'y a eu aucune réunion. Je vous rappelle
que nos statuts prévoient une session au moins tous les
six mois. Pour ce motif, je demande au conseil
d'administration de réagir et d'organiser une assemblée
immédiatement.
\closing{Dans l'attente d'une convocation}
\ps Ps: Ne voyez aucune animosité dans la présente.
\setkomavar*{enclseparator}{boîtier}
\enc{Extrait des statuts qui régissent les réunions
des membres.}
\cc{Le président du club\\tous les membres}
\end{letter}
\end{document}

```

Comme le montre la figure 3.20 page suivante, le résultat ne change pas.

Noter que dans l'exemple en allemand, le compte bancaire utilise « ß » produit par \tex-notation « \ss ». La raison en est que pendant le chargement de la classe, aucun paq pour le codage d'entrée, tel que `\usepackage[utf8]{inputenc}` pour les éditeurs modernes, ni pour changer la langue, par exemple pour la nouvelle orthographe allemande avec `\usepackage[ngerman]{babel}` n'est chargé.

Dans la table 3.10, vous trouvez une liste de tous les fichiers `.ico` prédéfinis. Si vous utilisez une imprimante qui a de grandes zones non imprimables sur la gauche ou sur la droite, vous pouvez avoir des problèmes avec l'option de `SN`. La norme `SN` suisse 101 130 prévoit que le champ d'adresse est placé à 8 mm du bord droit de la feuille, le titre et l'expéditeur seront mis avec la même petite distance du bord du papier à lettre suisse. Cela concerne aussi la ligne d'affaire lors de la création de `refline=wide` (large) (voir section 4.10).

Si vous avez un problème avec, créer votre propre fichier `lco` qui charge `SN` d'abord puis est ensuite placé dans les `toaddrhpos` (voir section 21.1.3) avec une valeur inférieure, diminuer `toaddrwidth` également en conséquence.

Le fichier `.ico` DIN est toujours chargé automatiquement le premier de sorte que toutes les pseudo-longueurs ont des valeurs par défaut raisonnables. Il n'est pas possible d'utiliser `\PassOptionsToPackage` pour passer des options dans un fichier `Ico` qui a déjà été chargé par la classe. Normalement, cela ne s'applique qu'à `typearea`, `scrfile`, `scrbase`, et `keyval`.

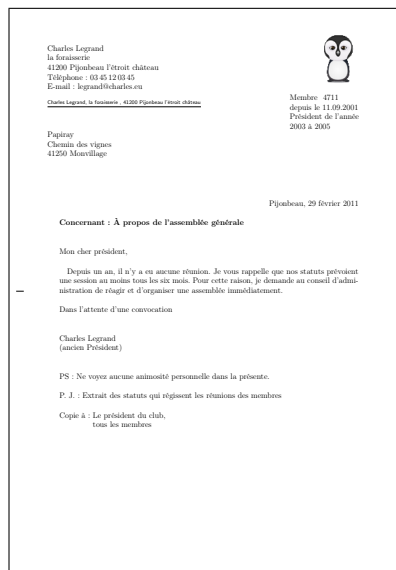


FIGURE 3.20 – Résultat d’une lettre avec ouverture, logo, adresse, date, sujet, titre, texte, fermeture, signature, post-scriptum, aménagements, liste de diffusion et marque de perforation avec un fichier `.ico`

TABLE 3.18 – Fichiers `.ico` prédéfinis

Éléments	Description
DIN	Ensemble des paramètres conformes à la norme DIN 676 pour enveloppes à fenêtres aux formats C4, C5, C6 et C6/5 (C6 long) pour lettres A4.
DINmtext	Ensemble des paramètres pour des lettres sur papier de format A4, conformes à la norme DIN 676, mais en utilisant une autre mise en page avec plus de texte sur la première page; ne convient que pour des enveloppes à fenêtre dans les tailles C6 et C6 / 5 (C6 long).

KakuLL	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Kaku A4, dans laquelle la fenêtre mesure 90 mm par 45 mm de haut, située à 25 mm de la gauche et 24 mm du bord supérieur (consulter l'annexe des instructions <b>KOMA-Script</b> en anglais).
KOMAold	Ensemble des paramètres pour des lettres au format A4 avec une disposition proche de la classe de lettre de <b>scrlettr</b> désormais obsolète, approprié pour des enveloppes à fenêtre dans les tailles C4, C5, C6 et C6 / 5 (C6 long) ; certaines commandes supplémentaires sont définies pour améliorer la compatibilité avec les commandes obsolètes de <b>scrlettr</b> ; <b>scrlttr2</b> peut se comporter d'une manière un peu différents lorsqu'il est utilisé avec ce fichier <b>lco</b> .
NF	Jeu de paramètres pour lettres françaises, selon la norme NF Z 11-001, adapté pour des enveloppes de type fenêtre au format lettre DL (110 mm à 220 mm) avec une fenêtre de 45 mm de largeur et 100 mm de hauteur à environ 20 mm du bord inférieur droit. Ce fichier a été développé, à l'origine, par Jean-Marie Pacquet qui fournit une version étendue, des informations complémentaires et une intégration LyX sur [ <b>Pac</b> ].
NipponEH	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Chou ou You 3 ou 4, dans lesquelles la fenêtre est d'environ 90 mm de large, 55 mm de hauteur, à 22 mm de la gauche et à 12 mm du bord supérieur (voir l'instruction <b>KOMA-Script</b> de l'annexe en anglais)
NipponEL	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Chou ou You 3 ou 4, dans lesquelles la fenêtre est d'environ 90 mm de large, 45 mm de hauteur, à 22 mm de la gauche et à 12 mm du bord supérieur (voir l'instruction <b>KOMA-Script</b> de l'annexe en anglais)

NipponLH	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Chou ou You 3 ou 4, dans lesquelles la fenêtre est d'environ 90 mm de large, 55 mm de hauteur, à 25 mm de la gauche et à 12 mm du bord supérieur (voir l' instruction KOMA-Script de l'annexe en anglais)
NipponLL	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Chou ou You 3 ou 4, dans lesquelles la fenêtre est d'environ 90 mm de large, 45 mm de hauteur, à 25 mm de la gauche et à 12 mm du bord supérieur (voir l' instruction KOMA-Script de l'annexe en anglais)
NipponRL	Jeu de paramètres pour lettres japonaises au format A4, adapté aux enveloppes à fenêtre de type Chou ou You 3 ou 4, dans lesquelles la fenêtre est d'environ 90 mm de large, 45 mm de hauteur, à 22 mm de la droite et à 28 mm du bord supérieur (voir l' instruction KOMA-Script de l'annexe en anglais)
SN	Jeu de paramètres pour lettres suisses, champ d'adresse sur le côté droit, selon la norme SN 010 130, adapté à la fenêtre dans les enveloppes de tailles C4, C5, C6 et C6/5 (C6 long).
SNleft	Jeu de paramètres pour lettres suisses, champ d'adresse sur le côté gauche adapté aux enveloppes à fenêtre avec fenêtre sur le côté gauche dans les tailles C4, C5, C6 et C6/5 (C6 long).
UScommercial9	Jeu de paramètres pour lettres États-Unis d'Amérique en format adapté aux enveloppes à fenêtre commerciales de taille num. 9 avec fenêtre de largeur 4 1/2, de hauteur 1 1/8 in, et une position de sept huitièmes de la gauche et de moitié par le bas, sans adresse de retour de l'expéditeur à l'intérieur de la fenêtre avec premier pliage sur la marque centrale. Le pli supérieur du papier peut également être utilisé dans le format légal, mais conduit à un avertissement sur le format du papier

`UScommercial9DW` Jeu de paramètres pour lettres Etats-Unis d'Amérique en format adapté aux enveloppes à fenêtre commerciales de taille num. 9 avec une fenêtre de largeur  $3 \frac{5}{8}$ , une hauteur de  $1 \frac{1}{8}$ , une position de  $\frac{3}{4}$  inch à partir de la moitié gauche et de  $\frac{1}{2}$  inch du bas de la fenêtre, avec une largeur de l'expéditeur de  $3 \frac{1}{2}$  et une hauteur de  $\frac{7}{8}$  inch à une position de  $\frac{5}{16}$  de la gauche et  $2 \frac{1}{2}$  à partir du bas, mais sans adresse de retour dans la fenêtre et premier pliage sur la marque centrale. Le pli supérieur du papier peut également être utilisé dans le format légal, mais conduit à un avertissement sur le format du papier

## 3.21 Adresse de fichiers et lettres circulaires

Lorsque l'on écrit des lettres circulaires l'une des tâches les plus fastidieuses est la saisie de nombreuses adresses différentes. La classe `sclttr2` offre un soutien de base pour cette tâche.

```
\adrentry{nom}{prénom}{adresse}{tél.}{F1}{F2}{commentaire}{clé}
```

La classe `sclttr2` prend en charge l'utilisation de fichiers d'adresses qui incluent les entrées très utiles pour les circulaires, avec extension `.adr`. Chaque entrée est une commande `\adrentry` avec huit paramètres, par exemple :

```
\adrentry{Lepetit}
{Français}
{3, chemin du bois \\41220 Tenetour}
{03\,45\,46 / 78\,90\,01}
{entrepreneur en bâtiment}
{}
{achète tout}
{LEPETIT}
```

Les 5e et 6e éléments, F1 et F2, peuvent être déterminés librement, par exemple, pour le sexe, le grade académique, l'anniversaire, ou la date de l'adhésion au club. Pour éviter l'écrasement des instructions `TEX` ou `LATEX`, il est conseillé d'utiliser des raccourcis uniquement en lettres majuscules.

Si M. PEUGNET est l'un de vos partenaires d'affaires les plus importantes avec lequel vous maintenez une correspondance journalière et que vous trouvez ennuyeux de taper encore et encore son adresse, alors `scrlttr2` peut vous aider. Si tous vos partenaires d'affaires ont une entrée dans un fichier d'adresses `partenaire.adr`. et si vous devez répondre à nouveau à M. PEUGNET, vous pouvez économiser votre frappe comme suit :

```
\input{partenaire.adr}
\begin{letter}{\PEUGNET}
Votre correspondance d'aujourd'hui
...
\end{letter}
```

Votre système  $\text{T}_{\text{E}}\text{X}$  doit être configuré pour avoir accès au fichier d'adresses. Sans accès, les résultats de la commande `\input` renvoient une erreur.

Vous pouvez mettre votre fichier d'adresses dans le répertoire d'exécution de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , ou configurer votre système pour rechercher le fichier dans un répertoire spécialement dédié.

```
\adrenentry{nom}{prénom}{adresse}{téléphone}{F1}{F2}{F3}{F4}{clé}
```

Au fil des années, vous m'avez objecté que `\adrenentry` n'a que deux paramètres libres. Pour répondre à cette demande, j'ai créé une nouvelle commande appelée `\addrenentry` - noter le « d » supplémentaire - qui en prend quatre librement définissables. Et depuis,  $\text{T}_{\text{E}}\text{X}$  prend en charge au maximum neuf paramètres par commande. Hors cette différence, l'utilisation est la même que celle de `\adrenentry`.

Les deux commandes `\adrenentry` et `\addrenentry` peuvent être utilisées dans les fichiers `.adr`. Noter toutefois qu'il existe des paqs qui ne sont pas adaptés à l'utilisation de `\addrenentry`. Par exemple, le paq `adrconv` par Axel KIELHORN peut être utilisé pour créer des listes d'adresses à partir de fichiers `.adr`, mais n'a actuellement aucun support pour la commande `\addrenentry`. Le seul choix, dans ce cas, est de faire vous-même des extensions appropriées du paq. Outre la facilité d'accès aux adresses, les fichiers `.adr` peuvent être utilisés pour écrire des lettres circulaires, et ne nécessitent plus d'accès à un système de base de données complexe par  $\text{T}_{\text{E}}\text{X}$ .

Si vous appartenez à une société de pêche et que vous voulez inviter tous les membres à participer à la prochaine assemblée générale :

```
\documentclass{scrlttr2}
\usepackage[french]{babel}
```

```

\usepackage[utf8]{inputenc}
\begin{document}
\renewcommand{\adrentry}[8]{ %
  \begin{letter}{#2 #1 \\\#3}
    \opening{Cher membre du club,}
    La prochaine assemblée générale aura lieu samedi 19 septembre 2015,
    à partir de 15h:
    Salle polyvalente - maison commune de la Cornée du Der.
    Les sujets suivants seront abordés \dots ...
    \closing{Cordialement,}
  \end{letter}
}
\Input{membres.adr}
\end{document}

```

Avec un fichier `.adr` qui contient aussi la commande `\addrentry`, une définition auxiliaire est nécessaire avant de charger le fichier d'adresses :

```

\renewcommand*{\addrentry}[9]{%
  \adrentry{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#9}%
}

```

Dans cet exemple simple, le paramètre librement définissable supplémentaire n'est pas utilisé, et `\addrentry` est défini à l'aide de `\adrentry`. Grâce à une programmation supplémentaire, on peut laisser le contenu des lettres dépendant des données d'adresse. Pour cela, les paramètres libres de `\adrentry` et de `\addrentry` peuvent être utilisés.

Supposons maintenant que le 5ème paramètre de la commande `\adrentry` contienne le genre d'un membre (m/f), et le 6e l'abonnement qui n'a pas encore été payé par ce membre. Si vous souhaitez écrire un rappel plus personnel à chacun des membres, l'exemple suivant peut vous aider :

```

\renewcommand*{\adrentry}[8]{
  \ifdim #6pt>0pt\relax
  \% #6 est un montant (float) supérieur à 0.
  \% Les mauvais payeurs sont détectables .
  \begin{letter}{#2 #1 \\\#3}
    \if #5m \opening{Cher #2,} \fi
    \if #5f \opening{Chère #2,} \fi

```

Malheureusement, nous avons noté que vous n'avez pas encore payé votre cotisation pour l'année en cours. Nous pensons qu'il s'agit

d'un oubli de votre part et nous vous demandons d'en régulariser le montant, soit 6 euros, par retour de courrier.

```
\closing{Cordialement,}
\end{letter}
\fi
}
```

Comme vous pouvez le voir, le texte de la lettre est personnalisé en fonction des caractéristiques spécifiques du destinataire. Le nombre d'attributs limité par le nombre de paramètres libres, deux pour la commande `\adrentry` ou quatre pour la commande `\addrentry`, permet de créer des listes d'adresses et de téléphone à l'aide de fichiers `.adr`. Pour cela, le paq supplémentaire `adrconv` par Axel KIELHORN (voir [Kie10]) est nécessaire.

Ce paq contient les documents L<sup>A</sup>T<sub>E</sub>X interactifs qui aident à créer ces listes.

<pre>\adrchar{lettre initiale} \addrchar{lettre initiale}</pre>
---

Les fichiers d'adresses doivent être préalablement triés afin d'obtenir des « listes triées ». Il est recommandé d'insérer, à chaque nouvelle lettre initiale du nom de famille, une déclaration `\adrchar` avec, comme argument, cette lettre pour séparer les entrées triées. Les commandes `\adrchar` et `\addrchar` peuvent être utilisées comme séparateur. Ces commandes seront ignorées si les fichiers d'adresses sont utilisés dans `scrlttr2`.

Exemple : vous avez le fichier d'adresses suivant pour créer un répertoire :

```
{adrchar\index{adrchar}}{A}
{adrentry\index{adrentry}}{Ange}{Gabriel}
    {nuage 3 \\ 12345 Paradis}
    {000\, 01\, 02\, 03}{-}{-}{Erzengel}
    {GABRIEL}
{adrentry\index{adrentry}}{Ange}{Michael}
    {nuage 3a \\12345 Paradis}
    {000\, 01\, 02\, 04}{-}{-}{Erzengel}
    {MICHAEL}
{adrentry\index{adrentry}}{Ange}{Raphael}
    {nuage 3b \\12345 Paradis}
    {000\, 01\, 02\, 01}{-}{-}{archange}
    {RAPHAEL}
{adrchar\index{adrchar}}{K}
{adrentry\index{adrentry}}{Kohm}{Markus}
```



```
{Baron von Drais - route 66 \\
68535 Edingen - - - Neckarhausen}
{+ 49-62\, 03-1\, 02\, 02}{-}{-}
{pas un ange du tout}
{KOMA}
{adrchar\index{adrchar}}{L}
{adrentry\index{adrentry}}{Lucifer}{Démon}
{Derrière la Flamme 1\\
66666 Gouffre de l'enfer}
{-}{-}{-}{ange déchu sans téléphone}
{LUCIFER}
{adrchar\index{adrchar}}{P}
{adrentry\index{adrentry}}{Papi}{Ray}
{Au bord du Lac \\
51000 La Cornée du Der}
{-}{-}{-}{téléphone à l'eau}
{PAPIRAY}
```

Ce fichier d'adresses peut être traité avec `adrdi`.`tex` du paq `adrconv`.

La lettre dans l'entête de page est créée avec la commande `\adrchar`.

La définition peut être trouvée dans `adrdi`.`tex`.

Les anciennes versions ne connaissent que les commandes `\adrentry` et `\adrchar`.

Consulter la documentation pour plus de détails sur le paq de `adrconv`.

Vous devriez également trouver des informations pour savoir si la version actuelle de `adrconv` soutient les commandes `\addrentry` et `\addrchar`.

Le résultat devrait ressembler à ceci :

		<b>A</b>
<hr/>		
<b>Ange, Gabriel</b>		
Nuage 3		
12345 Paradis	GABRIEL	
(Archange)	000 01 02 03	
<b>Ange, Michael</b>		
Nuage 3a		
12345 Paradis	MICHAEL	
(Archange)	000 01 02 04	

# Chapitre 4

## Entête et pied de page `sclayer-scrpage`

Jusqu'en 2001, il existait la solution du paq `scrpage` puis, jusqu'à la version 3.11b de `KOMA-Script`, le paq `scrpage2` a été la méthode recommandée pour personnaliser les entête et pied de page au-delà des possibilités offertes par les styles de page `headings`, `myheadings`, `plain`, et `empty` des classes `standard` `KOMA-Script`; `scrpage` est devenu obsolète et en 2013, plus de dix ans plus tard, a finalement été retiré de la distribution régulière de `KOMA-Script`

En 2013 le paq `sclayer` est devenu un module de base de `KOMA-Script`. Il fournit un niveau de modèle et un système basé sur le style de page. Néanmoins, sa souplesse, et la complexité qui en résulte, peuvent être trop exigeantes à manipuler pour l'utilisateur moyen. Pour plus d'informations sur cette interface, consulter la partie II, chapitre 16.

Beaucoup d'utilisateurs sont déjà familiers avec les instructions de `scrpage2` liées à la maîtrise de `sclayer` également documenté dans ce chapitre, car lui aussi nécessaire à l'utilisation de `sclayer-scrpage` fournissant l'interface usager sur la base de `sclayer` grandement améliorée, et largement compatible avec `scrpage2`. Ainsi, toute personne déjà familiarisée avec `scrpage2` et n'ayant pas eu recours à l'utilisation impropre de commandes internes de `scrpage2`, devrait facilement adopter en remplacement `sclayer-scrpage`. La plupart des exemples couvrant `scrpage2` dans les livres de `LATEX` ou de ressources en ligne devraient également travailler avec `sclayer-scrpage` soit directement,

soit avec des modifications mineures du code à condition que ces dernières collent aux interfaces standard.

Outre les paqs ci-dessus, vous pouvez utiliser également `fancyhdr` (voir [vO00]) pour configurer entêtes et pieds de pages. Cependant, `fancyhdr` n'a pas de support pour plusieurs options de KOMA-Script, par exemple une mauvaise prise en charge de l'élément police (voir `\setkomafont`, `\addtokomafont`, et `\usekomafont` à la section 3.6) ou le format de numérotation configurable pour les entêtes dynamiques (voir numéros d'option et `\chaptermarkformat` dans la section 3.16). Par conséquent, si vous usez d'une classe KOMA-Script, le paq `sclayer-scrpage` est recommandé. Vous pouvez, bien sûr, employer ce paq avec les classes `standard` de L<sup>A</sup>T<sub>E</sub>X. Avec le paq `sclayer-scrpage`, plusieurs autres fonctions sont fournies, d'un intérêt mineur pour l'utilisateur moyen, et pour cette raison, elles sont développées partie II, à partir du chapitre 17. Vous pouvez les consulter si les options décrites dans la partie I ne suffisent pas.

### 4.1 Option de sélection précoce ou tardive

Ce qui est écrit dans la section 2.4 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 2.4, vous pouvez passer à la section 5.3.

Cette section décrit une caractéristique du programme présenté et concerne `sclayer-scrpage`, mais aussi d'autres paqs et classes KOMA-Script. Ainsi, l'utilisateur trouvera les informations sur un paq ou une classe dans cette section qui figure presque à l'identique dans plusieurs chapitres. Les utilisateurs qui ne sont pas intéressés par une étude plus approfondie d'un paq ou d'une classe mais désirent seulement obtenir une vue d'ensemble de KOMA-Script peuvent lire ce paragraphe dans l'un des chapitres, puis le négliger.

```
documentclass[liste des options]{KOMA-Script-classe}
usepackage[liste des options]{liste des paqs}
```

Dans L<sup>A</sup>T<sub>E</sub>X, l'utilisateur spécifie les options de classe sous la forme d'une liste de mots clés simples séparés par une virgule comme argument optionnel, sauf que ces options sont également transmises à toutes les classes et à tous les paqs qui les intègrent. De même, il est admis que les utilisateurs spécifient comme argument optionnel de `usepackage` les options de paqs sous la forme d'une simple liste de mots-clés séparés par des virgules.

KOMA-Script étend ces possibilités en intégrant une valeur supplémentaire qui peut prendre la forme de `option=valeur`, utilisable indistinctement avec `documentclass` et `usepackage` de KOMA-Script ou avec L<sup>A</sup>T<sub>E</sub>X comme cela est décrit par exemple dans [DGS+12] et [Tea05b].

Le réglage des options a cependant, avec `documentclass` et `usepackage`, un inconvénient qui doit être considéré : commande, longueur, compteur... peuvent facilement casser. Ainsi, l'utilisation d'une longueur L<sup>A</sup>T<sub>E</sub>X comme valeur d'une option entraîne dans ces instructions un message d'erreur pour reprendre le contrôle avant que la valeur de KOMA-Script ne soit adoptée. L'affectation de valeurs qui peuvent être réalisées avec des longueurs de L<sup>A</sup>T<sub>E</sub>X ou instructions `KOMAoptions` et `KOMAoption` est documentée ci-dessous.

```
KOMAoptions{liste d'options}
KOMAoption{option}{liste des valeurs}
```

KOMA-Script offre, avec l'instruction `KOMAoptions` la possibilité de changer la valeur de la plupart des options de classes et de paqs, même après le chargement de la classe ou du paq. Vous pouvez rectifier les valeurs qui seront récupérées par chacune des options appartenant à la liste pour un certain nombre d'options disponibles. Certaines ont aussi une valeur par défaut (eng. `DefaultValue`). Si on omet de renseigner une valeur, c'est la valeur par défaut qui est appliquée automatiquement.

Dans certains cas, elles peuvent contenir simultanément plusieurs valeurs, avec la possibilité d'assigner à KOMA-Script une option séquentielle d'un ensemble de valeurs. Les valeurs individuelles sont, dans la liste de valeurs, séparées les unes des autres par des virgules.

Si une valeur contient un signe égal ou une virgule, alors la valeur est mise entre accolades.

KOMA-Script utilise pour la réalisation de cette possibilité les commandes `FamilyOptions` et `FamilyOption`, voir dans la partie II, la section 11.3.

## 4.2 Hauteurs de l'entête et du pied de page

Les classes `standard` de L<sup>A</sup>T<sub>E</sub>X utilisent peu le pied de page dont ils ont inclus le contenu dans un `\mbox`, ce qui se traduit par un pied avec une seule ligne de texte et probablement la raison d'une hauteur de pied mal définie, bien que `\footskip` donne la distance entre la dernière ligne de base de la zone de texte et la ligne de base du pied de page. Toutefois, si le pied de page se compose de

plus d'une ligne de texte, il n'y a pas de déclaration précisant si cette longueur représente la distance de la première ou la dernière ligne de base du pied de page. Malgré le fait que l'entête de page des classes `standard` soit également mis dans une boîte horizontale considérée comme une seule ligne de texte,  $\LaTeX$  fournit une longueur d'ajustement de cette hauteur qui est nécessaire pour déterminer le début de la zone de texte.

<code>\footheight</code>
<code>\headheight</code>

Un paq `\footheight` est introduit par `scrlayer` comme nouvelle longueur similaire à `\headheight` du noyau  $\LaTeX$ . De plus `scrlayer-scrpage` utilise `\footskip` qui interprète la distance de la dernière ligne de base possible de la zone de texte à la première ligne du pied de page de base normale. Le paq `typearea` interprète `footheight` de la même façon et les options de hauteur de pied de `typearea` peuvent également être utilisées pour régler les valeurs de paq `scrlayer` et `scrlayer-scrpage`. (voir les options `footheight` et `footlines` de la section 2.6 ainsi que l'option `footinclude` de la même section).

Si vous n'utilisez pas le paq `typearea`, vous devrez configurer manuellement la hauteur de l'entête et du pied en utilisant, si nécessaire, des longueurs chiffrées, et au moins le paq `geometry` pour fournir les paramètres correspondants.

Si vous avez installé l'entête ou le pied avec une hauteur trop petite pour le contenu, `scrlayer-scrpage` s'efforcera de régler correctement les longueurs correspondantes. En outre, il vous avertira et donnera des informations sur les modifications et les réglages appropriés que vous pourrez réaliser vous-même. Les changements automatiques sont mis en œuvre aussitôt avec toutefois une réserve : aucune suppression systématique, même si une exigence de hauteur inférieure du contenu est détectée ultérieurement.

### 4.3 Personnaliser le texte

Ce qui a été écrit dans la section 3.6 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 3.6, vous pouvez aller directement à la section 4.4.

Bien que `scrlayer` ne définisse pas les styles de page spécifiques avec un contenu - les styles de page mentionnés `@everystyle@` et `empty` sont en effet d'abord vides et sans niveau désigné - il offre des options et des commandes pour conditionner le contenu disponible.

```

\automark[Plan de la section de droite]{Plan de la section de
gauche}
\automark*[Plan de la section de droite]{Plan de la section de
gauche}
\manualmark
automark
autooneside=simple interrupteur
manualmark

```

Dans la plupart des classes et des rubriques, le choix du style de page `headings` et `myheadings` détermine comment les titres sont générés : automatiquement ou manuellement. Cette discrimination a été abolie avec `scrpage2` et maintenant avec `scrlayer`. Au lieu de faire la distinction entre les rubriques automatiques et manuelles selon le style de page, deux commandes `\automark` et `\manualmark` ont été créées.

Avec `manualmark` la substitution est manuelle, le réglage automatique arrêté, les niveaux de plan à utiliser pour le réglage automatique sont définis avec `automark` et `automark*`. L'argument optionnel dans ce cas indique le niveau de structuration de la marge de droite, tandis que l'argument obligatoire décrit le niveau de la marge gauche. Dans chaque cas, le nom du niveau du document est spécifié comme argument, à savoir `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` ou `subparagraph`.

Normalement, le niveau supérieur doit définir la marge de gauche, tandis que le niveau inférieur doit être utilisé pour la marge de droite. Bien qu'elles ne soient pas obligatoires, ces conventions communes sont simplement utiles.

Il convient de noter que toutes les classes ne permettent pas à `Kolumnentitel` d'agir à tous les niveaux, par exemple les classes `standard` `Kolumnentitel` `part` ne seront jamais définies alors que les classes KOMA-Script soutiennent tous les niveaux.

La différence entre `automark` et `automark*` est que `automark` annule toutes les commandes précédentes pour régler automatiquement les marges, tandis que la version étoilée `automark*` modifie uniquement les actions pour les niveaux hiérarchiques spécifiés. On peut donc couvrir également les cas relativement complexes.

Outre les commandes, les deux options `manualmark` et `automark` permettent également de basculer directement entre les titres automatique et manuel en cours d'exécution, `automark` est toujours utilisé avec la déclaration `chapter` comme valeur par défaut.

```
\automark[section]{chapter}
```

et avec d'autres classes

```
\automark[subsection]{section}
```

En principe, dans le mode unilatéral, on ne désire généralement pas que le niveau inférieur affecte la marge de droite. C'est plutôt le niveau supérieur qui indique également la valeur par défaut et qui influe dans l'exemple en mode recto-verso seulement sur la marge gauche pour afficher le titre de toutes les pages en cours. Cette préférence correspond à une option active `autooneside`. Elle peut prendre les valeurs pour les interrupteurs simples qui sont présentées dans le tableau 2.5 dans la table 2.5 page 59. Si l'option est désactivée, elle agit en mode unilatéral comme paramètre obligatoire dans le titre des colonnes.

Le chargement du paq lui-même n'a pas d'effet sur l'opportunité de travailler avec des entêtes de colonnes au fonctionnement automatique ou non. Seule l'utilisation explicite des options `automark` ou `manualmark` ou de l'une des deux commandes `\automark` ou `\manualmark` crée ici des conditions claires.

L<sup>A</sup>T<sub>E</sub>X a toute une série d'instructions pour personnaliser le texte. En plus du choix de polices, il inclut des commandes pour sélectionner une taille de police ou l'alignement du texte, plus de détails sur les possibilités normalement définies dans [DGS+12], [Tea05b] et [Tea05a].

<pre>\setkomafont{<i>élément</i>}{<i>commande</i>} \addtokomafont{<i>élément</i>}{<i>commande</i>} \usekomafont{<i>élément</i>}</pre>
---

En suivant les instructions `\setkomafont` et `\addtokomafont` il est possible de définir les commandes qui modifient les caractéristiques d'un élément qui est commuté. Théoriquement toutes les instructions possibles seront utilisées comme commandes, y compris du texte littéral, mais vous devez les limiter à des déclarations commutées avec un seul attribut de police. Habituellement, ce sont des commandes comme `\normalfont`, `\rmfamily`, `\sffamily`, `\ttfamily`, `\mdseries`, `\bfseries`, `\upshape` ou bien `\itshape`, `\slshape`, `\scshape` et encore les commandes `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize`, `\small`, `\footnotesize`, `\scriptsize` et `\tiny`. Pour en connaître le détail, consulter [DGS+12], [Tea05b] ou [Tea05a]. La substitution de couleur comme `\normalcolor` est possible (voir [Car05] et [Ker07]). L'utilisation d'autres commandes, spécialement celles qui permettent une redéfinition ou provoquent la sortie ne sont pas fournies. Un comportement étrange qui ne constitue pas un bug est possible dans ce cas.



Un élément de commutation de la police est `\setkomafont` fourni avec une toute nouvelle définition qui est seulement étendue en revanche avec la définition existante `\addtokomafont`. Attention, n'utilisez pas les deux déclarations dans le même document, mais seulement dans son préambule. Pour des exemples de l'utilisation, se référer aux sections sur les différents éléments. Les noms et significations des éléments individuels, ainsi que leurs paramètres par défaut, sont répertoriés dans la table 4.1 page suivante.

Avec la déclaration `\usekomafont` la police en cours peut être changée pour celle qui est définie pour l'élément spécifié.

<pre>\usefontofkomafont{élément} \useencodingofkomafont{élément} \usesizeofkomafont{élément} \usefamilyofkomafont{élément} \useseriesofkomafont{élément} \useshapeofkomafont{élément}</pre>
---

Parfois, dans la mise en écriture d'un élément, les choses qui sont faites n'ont rien à voir avec l'écriture, même si cela est expressément déconseillé.

Si seul le réglage de la police est effectué mais qu'aucun de ces paramètres supplémentaires ne sont exécutés, la déclaration `\usekomafont` est utilisée à la place de `\usefontofkomafont`.

Cette déclaration accepte uniquement la taille de la police et de la distance de référence, le codage (eng. *encoding*), la famille (eng. *family*), l'épaisseur de ligne ou la mise en forme (eng. *font series*) et la forme ou l'orientation (eng. *fontshape*).

Les instructions restantes concernant les attributs individuels de police peuvent être adoptées. Ici `usesizeofkomafont` accepte à la fois les tailles de la police et de la distance de base.

Ces commandes ne doivent pas être interprétées comme une légitimation pour toutes les instructions d'insertion dans les paramètres de la police des éléments. Cela peut très rapidement conduire à des erreurs (voir section 21.3).

## 4.4 Utilisation de pages avec styles prédéfinis

La meilleure façon d'obtenir, avec `sclayer-scrpage`, sa propre conception d'entête et de pied de page repose sur l'utilisation d'un style de page prédéfini.

#### 4 Entête et pied de page `sclayer-scrpage`

---

TABLE 4.1 – Éléments de `sclayer-scrpage` dont le type de style peut être changé avec `KOMA-Script` et paramètres par défaut si ceux-ci ne sont pas définis avant le chargement de `sclayer-scrpage`

---

Éléments
<b>footbotline</b>
Ligne en dessous du pied de page d'un style de page <code>sclayer-scrpage</code> . La police sera utilisée après <code>\normalfont</code> pour les éléments <code>pageheadfoot</code> et <code>pagefoot</code> . Il est recommandé de ne l'employer que pour les changements de couleur - vide par défaut.
<b>footsepline</b>
Ligne au-dessus du pied de page d'un style de page <code>sclayer-scrpage</code> . La police sera utilisée après <code>\normalfont</code> pour les éléments <code>pageheadfoot</code> et <code>pagefoot</code> . Il est recommandé de n'employer cet élément que pour changer de couleur - vide par défaut.
<b>headsepline</b>
Ligne en dessous de l'entête d'un style de page <code>sclayer-scrpage</code> . La police sera utilisée après <code>\normalfont</code> pour les éléments <code>pageheadfoot</code> et <code>pagefoot</code> . Il est recommandé de n'employer cet élément que pour les changements de couleur - vide par défaut.
<b>headtopline</b>
Ligne au-dessus de l'entête d'un style de page <code>sclayer-scrpage</code> . La police sera utilisée après <code>\normalfont</code> pour les éléments <code>pageheadfoot</code> et <code>pagefoot</code> . Il est recommandé de n'employer cet élément que pour les changements de couleur. - vide par défaut
<b>pagefoot</b>
Contenu du pied de page d'un style <code>sclayer-scrpage</code> , police utilisée après <code>\normalfont</code> pour l'élément <code>pageheadfoot</code> - vide par défaut.
<b>pagehead</b>
Contenu de l'entête d'un style <code>sclayer-scrpage</code> . La police sera utilisée après <code>\normalfont</code> pour l'élément <code>pageheadfoot</code> - vide par défaut.
<b>pageheadfoot</b>
Contenu de l'entête ou du pied d'un style de page que <code>sclayer-scrpage</code> définit avec <code>police</code> par défaut après <code>\normalfont</code> : <code>\NormalColor \slshape</code> .
<b>pagenumber</b>
Pagination réglée avec <code>\pagemark</code> . Si vous redéfinissez <code>\pagemark</code> , vous devez prendre soin que votre redéfinition utilise aussi <code>\usekomafontpagenumber</code> - par défaut : <code>\normalfont</code> .

---

<code>\scrheadings</code> <code>\plain.scrheadings</code>
--

Le paq `scrlayer-scrpage` offre deux styles de page à reconfigurer en fonction de vos propres souhaits.

Tout d’abord, le style de page `scrheadings` conçu pour être utilisé avec les titres courants. Ses défauts sont similaires aux rubriques de style de page `headings` de  $\LaTeX$  ou des classes `KOMA-Script`. Le contenu de l’entête et du pied de page peut être réglé en utilisant les commandes et les options décrites ci-dessous.

Deuxièmement, le style de page `plain.scrheadings` prévu pour être utilisé sans entête de colonne préconçue. Ses défauts sont similaires au style de page `plain` des classes `standard` de  $\LaTeX$  ou des classes `KOMA-Script`. Les commandes et les options utilisées pour mettre en forme le contenu de l’entête et du pied de page sont décrites ci-après.

Naturellement, vous pouvez configurer `scrheadings` comme style de page en cours d’exécution sans entête et `plain.scrheadings` comme style de page avec entête. Toutefois, il est conseillé d’adhérer à la convention sus-mentionnée surtout parce que les deux styles de page s’influencent mutuellement. Si vous sélectionnez l’un de ces styles, `scrheadings` reste accessible en tant que nom de rubrique de style et `plain.scrheadings` sera accessible aussi simplement.

Donc, si vous avez une classe ou un paq à basculer entre `headings` et `plain`, vous devez sélectionner `scrheadings` ou `plain.scrheadings` une seule fois et la classe ou le paq bascule automatiquement sans tenir compte du style de page. Aucun correctif ou adaptation de classes (ou de paq) ne sera nécessaire. Ainsi, ces deux styles de page peuvent être utilisés, sans préalable, en remplacement de `headings` et de `plain`. De plus amples informations dans l’article 17.2 de la partie II. Pour les utilisateurs `scrpage2` de l’ancien paq `KOMA-Script`, il est souligné que `plain.scrheadings` peut être traité pour des raisons de compatibilité avec ce paq sous le nom de l’alias `scrplain`.

Le contenu de l’entête des styles de page `plain.scrheadings` et `scrheadings` peut être paramétré avec les commandes ci-après. Ainsi l’argument facultatif définit le contenu d’un élément du style de page `plain.scrheadings`, tandis que l’argument obligatoire définit le contenu de l’élément correspondant du style de page `scrheadings`.

Le contenu de la page de gauche, donc côté pair que l’on appelle aussi « even », utilise les commandes de réglage `\lehead`, `\cehead` et `\rehead`. Le « e » à la deuxième place du nom de la commande signifie « even » dans le sens de pair.

```

\lehead[contenu plain.scrheadings]{contenu scrheadings}
\cehead[contenu plain.scrheadings]{contenu scrheadings}
\rehead[contenu plain.scrheadings]{contenu scrheadings}
\lohead[contenu plain.scrheadings]{contenu scrheadings}
\cohead[contenu plain.scrheadings]{contenu scrheadings}
\rohead[contenu plain.scrheadings]{contenu scrheadings}

```

Le contenu de la page de droite, donc côté impair que l'on appelle aussi `odd`, utilise les commandes de réglage `\lohead`, `\cohead` et `\rohead`. Le « o » à la deuxième place du nom de la commande signifie « odd » dans le sens de impair. Il faut remarquer que seules les pages impaires sont prises en compte par  $\text{\LaTeX}$  dans une simple mise en page pour la numérotation de page.

Chaque entête d'un style de page comporte un jeu d'éléments justifiés à gauche avec `\lehead` ou `\lohead`. Remarque : Le « l », pour left, à la première position des noms des commandes signifie « aligné à gauche ».

De même, chaque entête d'un style de page comporte un jeu d'éléments centrés qui peut être défini avec `\cehead` ou `\cohead`. Le « c », pour centred, au début du nom de la commande signifie « centré ». Enfin, chaque entête d'un style de page comporte un jeu d'éléments justifiés à droite avec `\rehead` ou `\rohead`. Le « r », pour right, au début du nom de la commande signifie « aligné à droite ».

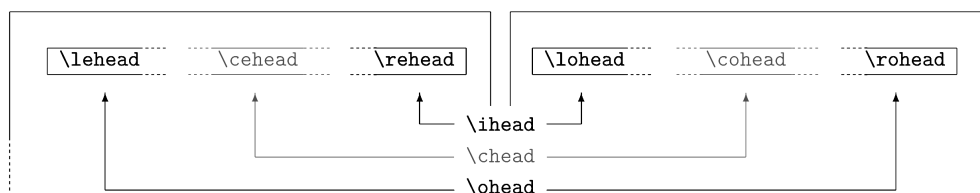


FIGURE 4.1 – Sémantique des commandes définissant l'entête d'un style de page `recto-verso`.

Chaque élément ne dispose pas de sa propre police qui peut être modifiée en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir [section 3.6](#)), et regroupées toutes ensemble sur l'élément `pagehead`. Avant cet ajout, la fonte de l'élément `pageheadfoot` sera utilisée. Les paramètres de ces deux éléments sont présentés dans la figure de la présente page qui montre l'importance des commandes individuelles pour l'entête des pages en mode `recto-verso` et pour la police par défaut de ces éléments.

Si vous écrivez un court article dont vous voulez afficher, dans l'entête de la page, le titre aligné à gauche et le nom de l'auteur aligné à droite, vous pouvez par exemple utiliser :

```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\pagestyle{scrheadings}
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\end{document}
```

Mais que se passe-t-il ? Sur la première page, seul le numéro de page apparaît dans le pied, l'entête reste vide ! L'explication est très simple. La classe de document `scrartcl` passe la classe `standard` sur le style de page `plain` par défaut. La commande `\pagestyle{scrheadings}` dans le préambule de notre exemple conduit à l'utilisation réelle du style de page `plain.scrheadings` pour la page avec entête.

L'utilisation d'une classe `KOMA-Script` par défaut pour ce style construit une page avec entête vide et un numéro de page dans le pied. Dans l'exemple du code utilisé, les arguments optionnels de `\lohead` et `\rohead` sont omis et le style de page `plain.scrheadings` reste inchangé par défaut. Le résultat en est donc, pour la première page, effectivement, correct. Puis ajouter, dans l'exemple de `\maketitle`, du texte jusqu'à imprimer une seconde page. Vous pouvez juste insérer `\usepackage{lipsum}` dans le préambule du document et `\lipsum` après `\maketitle`, le haut de la deuxième page contient maintenant, comme vous le souhaitez, le nom de l'auteur et le titre du document. Pour voir la différence, ajouter un argument optionnel de `\lohead` et `\rohead` avec un contenu. Pour ce faire, modifier l'exemple ci-dessus comme suit :

```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead[Papi Ray]
      {Papi Ray}
\rohead[Page style avec \KOMAScript]
      {Page style avec \KOMAScript}
\pagestyle{scrheadings}
```

```
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\end{document}
```

Maintenant, vous obtenez également un entête sur la première page au-dessus de l'entête de titre. Pour ce faire, vous avez simplement reconfiguré le style de page `plain.scrheadings` avec deux arguments optionnels. Mais reconnaissez qu'il serait préférable de laisser le style de cette page inchangé, parce que l'entête en cours d'exécution au-dessus du titre du document n'est probablement pas le meilleur choix. Alternative à la configuration `plain.scrheadings`, la classe `\titlepagestyle` de `KOMA-Script`, permet aussi de changer les titres d'entête, voir [section 3.12](#).

Permettez-moi une remarque importante : Vous ne devriez jamais mettre le titre ou le nombre de niveaux de la `section` directement dans l'entête de la page à l'aide d'une nouvelle déclaration par une de ces commandes.

```
\lehead*[contenu plain.scrheadings]{contenu scrheadings}
\cehead*[contenu plain.scrheadings]{contenu scrheadings}
\rehead*[contenu plain.scrheadings]{contenu scrheadings}
\lohead*[contenu plain.scrheadings]{contenu scrheadings}
\cohead*[contenu plain.scrheadings]{contenu scrheadings}
\rohead*[contenu plain.scrheadings]{contenu scrheadings}
```

En raison de l'asynchronisme de la mise en page et de la page de sortie de `TEX`, il peut arriver qu'un mauvais numéro ou un mauvais titre soit affiché dans l'entête en cours d'exécution. Au lieu de cela, le mécanisme `mark` devrait être utilisé en conjonction avec les automatismes décrits dans la `section` suivante. Ce qui est écrit dans le pied de page des styles `plain.scrheadings` et `scrheadings` est réglable à l'aide de ces commandes.

Un argument optionnel définit le contenu de l'élément de `plain.scrheadings` tandis qu'un argument obligatoire définit le contenu de l'élément correspondant du style de page `scrheadings` (voir à ce propos le chapitre 6.4 qui traite le problème des entêtes d'une manière identique).

```
\leftfoot[contenu plain.scrheadings]{contenu scrheadings}
\ceftfoot[contenu plain.scrheadings]{contenu scrheadings}
\reftfoot[contenu plain.scrheadings]{contenu scrheadings}
\loftfoot[contenu plain.scrheadings]{contenu scrheadings}
\coftfoot[contenu plain.scrheadings]{contenu scrheadings}
\roftfoot[contenu plain.scrheadings]{contenu scrheadings}
```

Comme les commandes d'entête, ces éléments ne disposent pas de leur propre police modifiable en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir section 3.6), et regroupées toutes ensemble sur l'élément `pagefoot`. Avant l'ajout, la fonte de l'élément `pageheadfoot` sera utilisée. Les paramètres qui montrent l'importance des commandes individuelles pour le pied de page en mode `recto-verso` et pour la police de ces éléments sont mis en évidence dans la figure de la présente page.

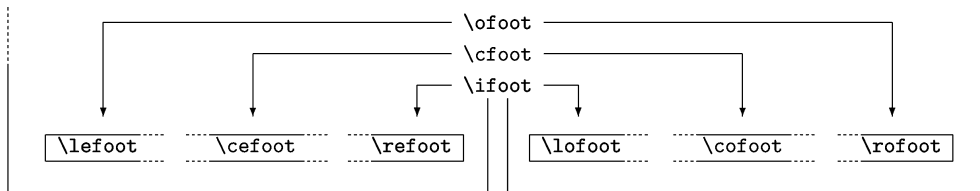


FIGURE 4.2 – Sémantique des commandes définissant le contenu du pied d'un style de page `recto-verso`.

Revenons à l'exemple de l'article court. Pour imprimer l'éditeur sur le côté gauche du pied de page, vous devez modifier l'exemple précédent comme suit :

```
\documentclass{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\lofoot{Editions La Cornée}
\pagestyle{scrheadings}
\usepackage{lipsum}
\begin{document}
\title{Page style avec \KOMAScript}
\author{Papi Ray}
\maketitle
\lipsum
\end{document}
```

Une fois de plus l'éditeur n'est pas imprimé sur la première page avec l'entête de titre, en voir l'explication sur `\lohead` dans l'exemple ci-dessus et la solution pour l'imprimer sur la première page serait similaire :

```
\lofoot[Editions La Cornée]{Editions La Cornée}
```

Maintenant, vous décidez de remplacer la police oblique dans l'entête et le pied droit par une police droite et plus petite, ce qui peut être réalisé en utilisant :

```
\setkomafont{pageheadfoot}{\small}
```

En outre, l'entête doit être en gras, mais pas le pied de page :

```
\setkomafont{pagehead}{\bfseries}
```

Pour cette dernière commande, il est important de ne la faire qu'après avoir chargé `scrpage-scrlayer` car la classe KOMA-Script définit déjà `pagehead` et `pageheadfoot` avec le même sens.

Le chargement de `scrpage-scrlayer` change le sens de `pagehead` et en fait un élément indépendant de `pageheadfoot`.

Vous pouvez maintenant compléter l'exemple `\lipsum` et même ajouter l'option `twoside` lors du chargement de `scrartcl`. Tout d'abord, le numéro de page dans le pied se déplace maintenant du centre vers l'extérieur.

Cela est dû à la valeur par défaut qui change pour `plain.scrheadings` et `scrheadings` avec une classe de document KOMA-Script `recto-verso`.

Simultanément, l'auteur, le titre du document et l'éditeur disparaissent de la page 2 et apparaissent seulement page 3, conséquence de l'utilisation des commandes pour les pages impaires. Vous pouvez les reconnaître par le « o » (`lofoot`, `cofoot`, `rofoot`) en deuxième position dans le nom des commandes. Nous pourrions tout simplement copier ces commandes et remplacer le « o » par un « e » pour définir le contenu des pages paires. Mais avec la mise en page `recto-verso`, il est plus logique d'utiliser des éléments de miroir inversé. Donc, l'élément de gauche d'une page impaire doit devenir l'élément de droite de la page paire, et vice versa. Pour ce faire, nous remplaçons aussi la première lettre « l » par « r » :

```
\documentclass[twoside]{scrartcl}
\usepackage{scrlayer-scrpage}
\lohead{Papi Ray}
\rohead{Page style avec \KOMAScript}
\lofoot[Editions La Cornée]
\rofoot[Editions La Cornée]
```



```

\rehead{Papi Ray}
\lohead{Page style avec \KOMAScript}
\refoot[Editions La Cornée]
{Editions La Cornée}
\pagestyle{scrheadings} \usepackage{lipsum}
\begin{document}
\title{Page styles avec \KOMAScript} \author{Papi Ray}
\maketitle
\lipsum
\lipsum
\end{document}

```

Après lecture de l'exemple, il peut vous sembler un peu lourd de dupliquer les commandes pour avoir le même contenu d'entête ou de pied de page en miroir sur une double page. Il existe une solution facile à mettre en place pour ce scénario tel que présenté ci-dessous.

```

\ohead[contenu plain.scrheadings]{contenu scrheadings}
\chead[contenu plain.scrheadings]{contenu scrheadings}
\rhead[contenu plain.scrheadings]{contenu scrheadings}
\ofoot[contenu plain.scrheadings]{contenu scrheadings}
\cfoot[contenu plain.scrheadings]{contenu scrheadings}
\ifoot[contenu plain.scrheadings]{contenu scrheadings}

```

Pour définir le contenu de l'entête et du pied des pages paires et impaires dans une présentation recto-verso, utiliser les commandes décrites ci-dessus. La teneur de la page paire sera définie à partir de la page impaire. Avec les pages souvent symétriques, un élément aligné à gauche sur une page paire est aligné à droite sur une page impaire et vice versa. Les éléments centrés sur les pages impaires le sont également sur les pages paires. Pour simplifier la conception des pages, `scrlayer-scrpage` utilise des abréviations. La commande `\ohead` équivaut à l'appel de `\lehead` et de `\rohead`, la commande `\chead` correspond à la fois à `\cehead` et à `\cohead`, et la commande `\ihead` à `\rehead` et à `\lohead`. Cela vaut également pour les instructions du pied. Les relations des entêtes et pieds de pages sont dans les figures 4.1 et 4.2. De nouvelles commandes permettent de simplifier l'exemple comme suit :

```

\documentclass[twoside]{scrartcl}
\usepackage{scrlayer-scrpage}
\ihead{Papi Ray}
\ohead{Page style with \KOMAScript}

```

```
\ifoot[Editions La Cornée]
{Editions La Cornée}
\pagestyle{scrheadings}
\usepackage{lipsum}
\begin{document}
\title{Page styles avec \KOMAScript}
\author{Papi Ray}
\maketitle
\lipsum
\end{document}
```

Comme vous pouvez le voir, en économisant la moitié des commandes vous obtenez le même résultat.

Dans les présentations recto, toutes les pages sont les pages impaires. Donc, en mode recto seul, les commandes  $\LaTeX$  sont synonymes de commandes de pages impaires. Par conséquent, dans la plupart des cas, vous aurez seulement besoin de ces six commandes au lieu des douze décrites précédemment.

`pagestyleset=réglage`

Dans les exemples ci-dessus, vous pouvez trouver quelques informations sur les valeurs par défaut de `scrheadings` et `plain.scrheadings`.

En effet `sclayer-scrpage` fournit deux pré-réglages différents d'origine. Vous pouvez choisir l'une de ces valeurs en utilisant l'option `pagestyleset` manuellement.

Avec `KOMA-Script` les valeurs par défaut sont utilisées et seraient également activées automatiquement si une classe `KOMA-Script` est détectée. Dans ce cas, et dans la mise en page `recto-verso`, `scrheadings` utilise la gestion de titres extérieurs alignés dans l'entête de la page courante. Le numéro de page sera imprimé aligné dans le pied de page. Dans le cas d'un seul côté le titre de la colonne est centré dans l'entête et le numéro de page centré dans le pied. Les majuscules et des minuscules seront automatiquement utilisées pour le titre de l'entête, ce qui correspond à l'utilisation de l'option simultanée `markcase=used`. La `pagestyle` de `plain.scrheadings` ne gère pas le paramétrage de l'entête, mais les numéros de page qui seront imprimés à l'identique.

Si le réglage est standard, les valeurs utilisées par défaut sont similaires aux rubriques de style de page `headings` et `plain` des classes `standard`. Ce paramètre sera également activé automatiquement si l'option n'a pas été utilisée et que les classes `KOMA-Script` ne sont pas détectées. Dans la mise en page `recto-verso`, `scrheadings` utilise les titres courants alignés en interne dans l'entête et les nu-

méros imprimés, également dans l'entête, alignés à extérieur. La mise en page simple face est identique, mais l'entête sera alignée à gauche et le numéro de page à droite.

L'entête en cours d'exécution automatique sera convertie en majuscules comme le ferait `markcase=upper` en dépit d'importantes considérations typographiques. L'utilisation du style de page `plain.scrheadings` diverge de `scrheadings` dans la mesure où le numéro de page s'imprime, en `recto-verso`, dans le milieu du pied. Noter que l'utilisation de cette option est activée en même temps que la page de style `scrheadings`. Ceci vaut également pour l'option utilisée dans un document. Les options `komastyle` et `standardstyle` existent encore pour `scrpage2` dans `sclayer-scrpage` pour une raison de compatibilité. Elles sont considérées comme obsolètes et ne devrait plus être utilisées.

## 4.5 Manipulation des styles de page prédéfinis

Nous avons expliqué comment sont prédéfinis et comment les valeurs des styles de page `plain.scrheadings` et `scrheadings` peuvent être modifiées. Cependant, il manque encore des informations sur, par exemple, la génération d'un titre courant, ou bien comment changer la largeur de l'entête et du pied, ou encore sur la façon de mettre des lignes au-dessus de l'entête ou en dessous du pied de page. Bien que ces fonctionnalités appartiennent au paq `sclayer`, elles sont expliquées ci-dessous, les propriétés de base de `sclayer` sont une partie importante des possibilités de `sclayer-scrpage`.

```
\automark[Plan de la section de droite] {Plan de la section de
gauche}
\automark*[Plan de la section de droite] {Plan de la section de gauche}
\manualmark
automark
autooneside=simple interrupteur
manualmark
```

Avec les classes `standard` de `LATEX` et de `KOMA-Script`, vous pouvez décider d'adopter des entêtes au fonctionnement automatique ou avec réglage manuel en utilisant un style de page `headings` ou `myheadings` approprié. Comme il est écrit dans l'article 3.12, les entêtes sont obtenus en fonction des titres en cours d'exécution. Les titres courants automatiques sont souvent la réplique d'un extrait de texte important caractéristique de la page, la plupart du temps à l'intérieur de l'entête, parfois, mais plus rarement, dans le pied.

La classe `scrartcl` avec le style de page `headings` utilise le titre de `section` en cours ou l'argument optionnel de `\section`, comme entête de documents dans le mode simple page. Ceci est considéré comme `section` par défaut. Les documents `recto-verso` utilisent cette position intitulée `section` de gauche tandis que dans le même temps, le sous-titre de `section` utilise une position de paragraphe à droite. La `section` de gauche sera imprimée sur les pages impaires, de gauche, d'où l'origine de `leftmark`, la `section` de droite sur les pages paires de droite, et en mode unilatéral, sur toutes les pages. Les classes par défaut éliminent la page paire à chaque fois que l'entête de `section` est dans une page impaire.

Les classes `report` et `book` commencent à un niveau plus élevé. Elles utilisent le titre du chapitre pour la `section` de droite dans une mise en page à un seul côté. Dans le mode `recto-verso`, la mise en page utilise le titre du chapitre pour la `section` de gauche et le titre de la `section` pour la `section` de droite.

Si vous utilisez `myheadings` comme style de page, les sections dans l'entête existent toujours et seront imprimées, ainsi que les numéros de page même si elles n'ont pas été automatiquement fixées par les rubriques. Ainsi, vous pouvez les remplir manuellement en utilisant uniquement les commandes `\markright`, `\markleft`, et `\markboth`, documentées dans cette `section`.

Cette différence entre les deux styles de page a été supprimée par `scrpage2` et par `scrlayer`. Au lieu de faire la distinction entre les entêtes automatique et manuel par le choix d'un style de page, deux nouvelles commandes, `\automark` et `\manualmark`, sont fournies. La commande `\manualmark` passe aux sections manuelles et désactive le paramètre automatique, contrairement à `\automark` et `\automark*` et peut être employée pour définir les niveaux de `section` qui doivent être utilisés pour le réglage automatique. L'argument optionnel indique le repère du niveau de l'organisation à droite, l'argument obligatoire le niveau de la `section` à gauche.

Les arguments doivent toujours porter le nom d'un niveau de section tel que `part` (partie), `chapter` (chapitre), `section`, `subsection` (sous-section), `subsubsection` (sous-sous-section), `paragraph`, mais aussi `subparagraph` (sous-paragraphe) ou `alinéa`.

En général, le niveau le plus élevé doit être utilisé pour la `section` de gauche et le niveau inférieur pour la `section` de droite. Il s'agit là d'une simple convention logique et non d'une nécessité.

Il est à noter que toutes les classes « titre de colonne » ne conviennent pas forcément à tous les niveaux, par exemple les classes `standard` ne les utilisent

jamais avec `\part`, mais que les classes KOMA-Script, cependant, soutiennent tous les niveaux.

La différence entre `\automark` et `\automark*` est que `\automark` annule toutes les commandes précédentes pour régler automatiquement les sections, tandis que la version étoilée `\automark*` ne modifie les actions que pour les niveaux de détails spécifiés. Vous pouvez ensuite construire des cas plus complexes.

Si vous voulez le titre du chapitre sur les pages de gauche d'un livre, les entêtes de chapitre sont utilisés comme titres de fonctionnement des sections sur ce côté au lieu d'être exécutés sur les pages impaires, comme il est habituel pour un livre. Toutefois, pour les pages de droite, vous souhaitez également les entêtes de chapitre aussi longtemps que le titre du chapitre puisse être utilisé, jusqu'à ce que la première `section` apparaisse. Pour ce faire, vous devez d'abord charger `scrlayer-scrpage` et sélectionner le style de page `scrheadings`. Ainsi, votre document commence par :

```
\documentclass{scrbook}
\package{scrlayer-scrpage}
\pagestyle{scrheadings}
```

Ensuite, veiller à ce que les entêtes de chapitre définissent à la fois les sections gauche et droite :

```
\automark[chapter]{chapter}
```

Puis les titres de `section` doivent également définir les bonnes sections :

```
\automark*[section]{}
```

Voici la version étoilée de l'application utilisée depuis la déclaration précédente `\automark` qui doit encore être valide. En outre, l'argument obligatoire pour le niveau de la `section` de gauche est laissé vide (`empty`) parce que cette `section` doit rester inchangée.

Maintenant, vous avez juste besoin d'un peu de contenu dans votre document pour voir le résultat :

```
\usepackage{lipsum}
\begin{document}
\chapter{chapitre}
\lipsum[1-20]
\section{section}
\lipsum[21-40]
\end{document}
```

Nous utilisons une fois encore le paq `lipsum` pour générer un texte bidon avec la commande `\lipsum`. Ce paq est vraiment utile.

Si vous testez l'exemple, vous verrez que la page chapitre de démarrage est sans titre, comme d'habitude, car ce dernier est automatiquement introduit dans la mise en page du style de page `plain` de `\plain.scrheadings`. Les pages 2-4 portent les titres des entêtes de chapitre : le titre de `section` page 4, changement du titre de la rubrique page 5. Dès lors, les deux titres sont affichés alternativement dans l'entête, sur le côté gauche de l'entête du chapitre et sur le côté droit de l'entête de la `section`.

Vous pouvez également utiliser les options `manualmark` et `automark` pour basculer entre les commandes manuelles et automatiques de fonctionnement des entêtes. Ainsi `automark` utilise toujours par défaut :

```
\automark[section]{chapter}
```

pour les classes avec `\chapter` et :

```
\automark[subsection]{section}
```

pour les classes sans `\chapter`.

Si, en mode unilatéral, vous ne voulez pas que le niveau inférieur influe sur la `section` de droite mais que, par exemple, seul le niveau supérieur en mode `recto-verso` affecte la `section` de gauche et indique les titres des entêtes de toutes les pages, l'option par défaut `autooneside` permet ce comportement. Elle comprend les valeurs pour les interrupteurs simples, qui peuvent être trouvées dans la table 1.6 page 20. Vous pouvez la désactiver pour une option de mise en page simple face, les arguments obligatoires de `\automark` et `\automark*` influenceront encore l'entête en cours d'exécution.

Pour avoir un rapport simple face avec un entête similaire à celui du livre dans l'exemple précédent. le titre du chapitre devrait être utilisé aussi longtemps que la première `section` apparaît. À cet effet, l'exemple est modifié comme suit :

```
\documentclass{scrreprt}
\usepackage[autooneside=false]{sclayer-scrpage}
\pagestyle{scrheadings}
\automark[section]{chapter}
\usepackage{lipsum}
\begin{document}
\chapter{Titre du chapitre}
\lipsum[1-20]
\section{Titre de la section}
```

```
\lipsum[21-40]
\end{document}
```

Dans ce cas, nous n'utilisons pas la commande `\automark*`. Vous pouvez essayer l'exemple avec `autooneside=true` ou supprimer l'option et sa valeur. Vous devriez trouver une différence dans l'entête de la page 4 en cours d'exécution. Noter que le chargement du paq n'a pas d'effet sur le fait de savoir si vous utilisez un entête automatique ou manuel, ou si ce genre de titre ne remplit pas les sections. Seule l'utilisation explicite des options `automark` ou `manualmark`, ou des commandes `\automark` ou `\manualmark` produit ici une situation claire. Si nécessaire, vous pouvez trouver des exemples d'utilisation de ces commandes et options avec `scrlayer` basé sur `scrlayer-scrpage` dans la section 4.5,

```
draft=ouvert-fermé-valeur
```

`KOMA-Script-Option` peut prendre les valeurs standard pour commutateurs simples, comme indiqué dans la table 2.5 page 59. Si l'option est activée, tous les éléments de styles de page sont en outre fournis avec des lignes de cote. Cela peut parfois être utile pendant la phase de conception.

```
\MakeMarkcase{texte}
```

L'exécution automatique des titres utilise `\MakeMarkcase` pour leur édition, mais pas le titre manuel des colonnes. Si l'instruction de chargement `scrlayer` n'est pas définie, l'argument réglage par défaut est utilisé et le texte inchangé. Ce défaut peut être modifié soit en redéfinissant `\MakeMarkcase` ou à l'aide de l'option documentée ci-dessous. Selon le réglage, l'argument est, par exemple, converti ensuite en majuscules ou en minuscules.

```
markcase=valeur
```

Comme expliqué plus haut, vous pouvez choisir avec `scrlayer` entre manuel et automatique pour les titres en cours d'exécution, avec la mise en forme automatique dans l'entête tandis que les marges appropriées seront définies avec les commandes de sectionnement.

Chaque culture obéit à ses propres règles de typographie, parfois différentes de celles appliquées dans les pays germanophones, telles le titre des rubriques en lettres majuscules<sup>1</sup>, permises par défaut dans les classes `standard` de `LATEX`. Le paq `scrlayer` le supporte également en option, avec `markcase=upper`. En fin de compte, cela conduit à une redéfinition de `\MakeMarkcase`, déclaration utilisée par `scrlayer` dans la mise en place du titre de l'entête en cours d'exécution.

1. Dans les indications de typographie allemande, aucune différence n'est signalée par rapport aux usages pratiqués pour des textes en français.

Pour contrôler cette fonction, vous devez utiliser l’option `markcase=upper` qui se traduit par une redéfinition de la commande `\MakeMarkcase` utilisée par `sclayer` pour le fonctionnement automatique des entêtes. Cependant, il faut reconnaître que les résultats de la commande de L<sup>A</sup>T<sub>E</sub>X, `\MakeUppercase`, qui propose la composition avec des lettres supérieures n’utilisant ni l’espace, ni l’équilibrage de ces lettres, ne sont pas satisfaisants.

Ils sont dûs en partie à l’interprétation typographiquement correcte exigeant une analyse de glyphe afin d’incorporer la forme spécifique des lettres et leurs combinaisons dans l’équilibre du bloc. L’auteur de `KOMA-Script` recommande de renoncer à cette application pour `Kolumnentitel`, habituellement utilisé avec `markcase=used`. Cependant, certaines classes ajoutent `MakeUppercase` ou même la commande T<sub>E</sub>X `\uppercase`.

Pour de tels cas, l’option `markcasenoupper` peut être utilisée. Cela permet également de désactiver `\MakeUppercase` et `\uppercase` dans l’entête local. Toutes les valeurs possibles pour `markcase` sont résumées dans la table 4.2.

TABLE 4.2 – Valeurs possibles de l’option `markcase` pour le choix des minuscules/majuscules dans la composition des entêtes de colonnes automatiques.

<code>lower</code>
redéfinit <code>\MakeMarkcase</code> pour convertir les titres courants automatiques en lettres minuscules avec <code>\MakeLowercase</code> (minuscules composition).
<code>upper</code>
redéfinit <code>\MakeMarkcase</code> pour convertir les titres courants automatiques en lettres majuscules avec <code>\MakeUppercase</code> (majuscules composition).
<code>used</code>
redéfinit <code>\MakeMarkcase</code> pour utiliser des entêtes de fonctionnement automatique, sans aucune modification de cas.
<code>ignoreuppercase, nouppercase, ignoreupper, noupper</code>
redéfinit <code>\MakeMarkcase</code> et empêche, pour le titre courant, d’effectuer un changement automatique minuscule/majuscule. Localement désactivé pour tous les niveaux de tous les styles de page des entêtes exécutant <code>\MakeUppercase</code> .



```
\leftmark  
\rightmark  
\headmark  
\pagemark
```

Pour déroger aux styles prédéfinis, il faut généralement être en mesure de décider où les repères doivent être placés. Avec `\leftmark` le numéro est à gauche. Il est ensuite remplacé lors de la sortie de la page avec le contenu approprié.

Vous pouvez donc utiliser `\rightmark` pour placer à droite le numéro qui sera ensuite remplacé par le contenu correspondant à la sortie de page.

Pour plus de subtilités vous trouverez des explications supplémentaires dans l'article 21.1.

Utiliser `\headmark` peut vous rendre la vie plus facile.

Cette extension de `sclayer` correspond à `\leftmark` ou `\rightmark` selon que la page en cours est à gauche ou à droite.

La commande `\pagemark` n'a strictement rien à voir avec les marques de  $\TeX$ . Elle sert à formater le numéro de page. Pour créer ce numéro et pour le réglage de l'élément de la police, `pagenumber` est utilisé. Cela peut être modifié en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir également la section 3.6) Vous trouverez un exemple d'utilisation des commandes `\headmark` et `\pagemark` avec `sclayer` basé sur le paq `sclayer-scrpage`, section 5.5.

Si les possibilités offertes pour les marques ne suffisent pas, des instructions sont documentées dans ce chapitre pour les utilisateurs avancés, comme par exemple l'usage de `\leftfirstmark` et de `\rightbotmark` expliqué pour les documents de type encyclopédie.

```
\partmarkformat  
\chaptermarkformat  
\sectionmarkformat  
\subsectionmarkformat  
\subsubsectionmarkformat  
\paragraphmarkformat  
\subparagraphmarkformat
```

Ces instructions sont utilisées en interne par `KOMA-Script` et `sclayer` pour formater le numéro automatique du titre de `section` en cours d'exécution. Ici aussi, le mécanisme `\autodot` des classes `KOMA-Script` est pris en charge. Si

nécessaire, elles peuvent être redéfinies afin de procéder à une autre mise en forme des numéros. Voir l'exemple dans la **section 5.5**.

```
\partmark{texte}  
\chaptermark{texte}  
\sectionmark{texte}  
\subsectionmark{texte}  
\subsubsectionmark{texte}  
\paragraphmark{texte}  
\subparagraphmark{texte}
```

Ces instructions, utilisées par la plupart des classes, permettent de définir en interne les sections en parlant des commandes de sectionnement. Dans ce cas, seul le texte sans le numéro est attendu comme un argument. Ce nombre est déterminé automatiquement à la place du numéro actuel, si vous travaillez avec des titres numérotés. Si ces instructions sont redéfinies par l'utilisateur, il doit, avant de fixer le nouveau nombre, impérativement vérifier sur **secnumdepth** les chiffres du niveau de la **section**, même dans le cas où il ne change pas lui-même **secnumdepth**, parce que les paqs et les classes peuvent le faire localement et compter sur une manipulation correcte de **secnumdepth**.

Cependant, de telles instructions ne sont pas utilisées dans toutes les classes, à tous les niveaux. Par exemple, `\partmark` n'est jamais appelé dans les classes **standard**, alors que les classes **KOMA-Script** le soutiennent. Le paq **sclayer** définit également ces instructions partiellement renouvelées après chaque appel à `\manualmark` ou `\automark` ou les options appropriées pour désactiver ou activer le titre de fonctionnement automatique ou manuel souhaité.

```
\markleft{section gauche}  
\markright{section droite}  
\markboth{section gauche}{section droite}
```

Indépendamment du fait que les titres soient manuels ou automatiques, vous pouvez modifier le contenu des sections gauche ou droite avec l'une de ces instructions à tout moment. Il est à noter que le repère de gauche généré dans `\leftmark` est la **section** de gauche de la dernière commande `\markboth` ou `\markleft` de la page en cours, tandis que le repère de droite généré dans `\rightmark` est la **section** droite de la première commande `\rightmark` ou `\markboth` de la page en cours. Plus de détails sur `\rightmark` **section 20.1**.

Lorsque vous travaillez avec des titres manuels, les sections restent valables jusqu'à ce qu'elles soient explicitement remplacées par une nouvelle instruction équivalente. Lors de l'exécution automatique de l'entête, elles peuvent perdre

leur validité avec la mutation de l'entête de `section` en fonction de la manière dont l'automatisme est configuré.

Toujours dans le contexte en relation avec les versions étoilées de commandes de sectionnement, ces instructions peuvent être utiles. Des exemples détaillés de l'utilisation de `\markboth` avec le paq `scrlayer` dérivé de `scrlayer-scrpage` peuvent être trouvés dans la section 5.5.

Supposons par exemple que vous écrivez juste avant la table des matières une préface de plusieurs pages qui ne doit pas apparaître. Les titres des colonnes doivent mettre la préface en évidence et vous voulez aussi un entête en cours d'exécution pour cette préface :

```
\documentclass[headsepline]{book}
\usepackage[automark]{scrlayer-scrpage}
\pagestyle{scrheadings}
\usepackage{blindtext}
\begin{document}
\chapter*{Preface}
\markboth{Preface}{Preface}
\blindtext[20]
\tableofcontents
\blinddocument
\end{document}
```

Dans un premier temps, cela semble produire le résultat escompté, mais un examen plus attentif permet de noter que la colonne intitulée `Préface` n'est pas, contrairement aux autres titres de colonnes, en lettres majuscules.

Vous pouvez facilement résoudre ce problème :

```
\documentclass[headsepline]{scrbook}
\usepackage[automark]{scrlayer-scrpage}
\pagestyle{scrheadings}
\usepackage{blindtext}
\begin{document}
\chapter*{Preface}
\markboth{\MakeMarkcase{Preface}}
           {\MakeMarkcase{Preface}}
\blindtext[20]
\tableofcontents
\blinddocument
\end{document}
```

Comme on peut le voir, la commande `\MakeMarkcase` est utilisée également pour ajuster le titre manuellement corrigé et mettre la préface en conformité avec les titres automatiques du reste du document. Maintenant, passons la commande `\tableofcontents` en face de la préface et enlevons la commande `\markboth`. Vous constatez que la préface porte maintenant comme titre courant « contents » en cours d'exécution. Ceci est dû à une particularité de `\chapter*` (voir section 3.16). Si, en pareil cas, l'entête en cours d'exécution doit être retirée, `\markboth` serait utile avec deux arguments vides :

```
\documentclass[headsepline]{book}
\usepackage[automark]{sclayer-scrpage}
\pagestyle{scrheadings}
\usepackage{blindtext}
\begin{document}
\tableofcontents
\chapter*{Preface}
\markboth{}{}
\blindtext[20]
\blinddocument
\end{document}
```

<code>headwidth=largeur: offset: offset</code>
<code>footwidth=largeur: offset: offset</code>

Par défaut, l'entête et le pied de la page ont la même largeur que la zone de texte. Cela peut être modifié en utilisant ces options **KOMA-Script**. La valeur de la largeur indiquée est la largeur voulue respectivement pour l'entête et le pied de page. Le décalage définit de combien l'entête ou le pied doivent être déplacés vers la marge extérieure, dans une mise en page simple, le côté droit. Les deux valeurs sont facultatives et peuvent être omises. Dans ce cas, vous pouvez également omettre le double point.

Une seule valeur sans les deux points indique la largeur.

Pour la largeur et le décalage, utiliser une valeur de longueur valide, longueur de  $\LaTeX$  et de  $\TeX$  ou espace de  $\TeX$ . En outre, vous pouvez utiliser une expression de dimension  $\varepsilon\text{-TeX}$  avec des opérations arithmétiques de base  $+$ ,  $-$ ,  $*$ ,  $/$  et les parenthèses. Voir [Tea98], section 3.5 pour plus d'informations sur ces expressions et la section 6.1 pour plus d'informations sur l'utilisation d'une longueur de  $\LaTeX$  comme option de valeur. La largeur peut alternativement être l'une des valeurs symboliques indiquées dans la table 4.3 page 272. Sa valeur par défaut correspond à la largeur de la zone de texte. La valeur par défaut

pour le décalage dépend de la largeur choisie. La moitié de la différence entre la largeur et la largeur de la zone de texte est, en général, utilisée dans un sens unique. Ainsi, l'entête au-dessus du champ de texte est centrée.

Dans une mise en page **recto-verso**, seul un tiers de la différence entre cette largeur et la largeur de la zone de texte est utilisée. La largeur est donc la largeur de l'ensemble de la zone de texte augmentée de la colonne de note marginale, et le décalage sera de zéro par défaut.

Pour éviter des calculs trop compliqués, il suffit simplement de spécifier le décalage désiré.

<code>headtopline=épaisseur: longueur</code>
<code>headsepline=épaisseur: longueur</code>
<code>footsepline=épaisseur: longueur</code>
<code>footbotline=épaisseur: longueur</code>

Les classes **KOMA-Script** fournissent deux lignes de séparation activables et désactivables uniquement en option, l'une en dessous de l'entête et l'autre au-dessus du pied de page.

Le paq **sclayer-scrpage** fournit quatre lignes horizontales : une au-dessus et une en dessous de l'entête, une au-dessus et une en dessous du pied. Chaque ligne peut être activée et configurée séparément en longueur et épaisseur.

Les deux valeurs sont en option. Si vous omettez l'épaisseur, une valeur par défaut, dite un « délié » de 0,4 pt sera utilisée. Si vous omettez la longueur, la largeur de l'entête respective sera utilisée pour le pied. Si vous omettez les deux, vous pouvez également ne pas mettre le double point. Si vous utilisez une seule valeur sans les deux points, ce sera l'épaisseur.

Bien entendu, la longueur doit être non seulement inférieure à la largeur courante de l'entête, mais aussi du pied. Pourtant, elle peut également être plus longue, voir les options **l**ines, **c**lines et **o**lines, expliquées plus loin dans cette section. En plus de l'épaisseur et de la longueur, la couleur des lignes peut aussi être modifiée, ce qui, tout d'abord, dépend bien sûr de la couleur définie pour l'entête et le pied.

Mais indépendamment du réglage par défaut, **headtopline**, **headsepline**, **footsepline** et **footbotline** seront utilisés en supplément pour les éléments correspondants. En utilisant `\setkomafont` ou `\adatokomafont`, vous pouvez les modifier (voir section 3.6). Par défaut ces paramètres sont vides, ce qui signifie que des changements de la police courante ou de la couleur qui ne seraient pas logiques ne sont pas recommandés pour ces éléments.

#### 4 Entête et pied de page *scrlayer-scrpage*

---

TABLE 4.3 – Valeurs symboliques autorisées pour les options de largeur d’entête et de pied de page

---

<b>foot</b>
la largeur actuelle du pied de page
<b>footbotline</b>
la longueur actuelle de la ligne horizontale au-dessous du pied de page
<b>footsepline</b>
la longueur actuelle de la ligne horizontale au-dessus du pied de page
<b>head</b>
la largeur actuelle de l’entête de la page
<b>headsepline</b>
la longueur actuelle de la ligne horizontale au-dessous de l’entête de la page
<b>headtopline</b>
la longueur actuelle de la ligne horizontale au-dessus de l’entête de la page
<b>marginpar</b>
la largeur actuelle de la colonne de note marginale y compris la distance entre la zone de texte et la colonne de la note dans la marge
<b>page</b>
la largeur de la page, en tenant compte d’une possible correction de la liaison définie en utilisant le paq <code>typearea</code> (en option <i>BCOR</i> voir section 2.6)
<b>paper</b>
la largeur actuelle du papier sans tenir compte d’une correction de liaison
<b>text</b>
la largeur actuelle de la zone de texte
<b>textwithmarginpar</b>
la largeur actuelle de la zone de texte et de la colonne marginale y compris la distance entre les deux (Attention : dans ce cas et seulement dans ce cas, la valeur par défaut pour compenser est zéro)

---

Les quatre plus importantes commandes `\setheadtopline`, `\setheadsepline`, `\setfootsepline` et `\setfootbotline` comportent des options correspondantes dans `scrpage2`, et comptent un premier argument optionnel de longueur, suivi d'un second argument obligatoire pour l'épaisseur et du troisième optionnel pour le réglage de police ou la couleur.

Ces instructions existent dans le paq `sclayer-scrpage` pour des raisons de compatibilité. Elles sont considérées comme obsolètes et ne devraient plus être utilisées. Pour être clair : ces commandes n'ont jamais été faites pour changer ou supprimer les lignes. Au contraire, elles étaient les seules à permettre d'ajuster la longueur et l'épaisseur de ces dernières, ce qui a été malheureusement souvent ignoré par les utilisateurs.

```
plainheadtopline=simple interrupteur
plainheadsepline=simple interrupteur
plainfootsepline=simple interrupteur
plainfootbotline=simple interrupteur
```

Avec ces options, les paramètres pour les lignes peuvent être appliqués pour le style simple de la page. La valeur marche-arrêt est la valeur standard disponible pour les interrupteurs simples, comme indiqué dans la table 1.6 page 20. Si cette option est activée, les paramètres sont appliqués. Lorsqu'elle est désactivée, le style simple page, `plain`, ne sera pas affiché sur la ligne correspondante.

```
ilines
clines
olines
```

Nous avons déjà vu qu'une ligne horizontale au-dessus ou en dessous du pied de page ou de l'entête peut être plus longue ou plus courte que cet entête ou ce pied de page. Seule la réponse à la question sur l'alignement de cette ligne est toujours manquante. Par défaut, les lignes sont alignées du côté gauche dans la mise en page simple face et justifiées sur la marge intérieure de l'entête ou du pied dans la mise en page `recto-verso`. Ceci correspond à l'option `ilines`.

Alternativement, vous pouvez utiliser l'option `clines` pour centrer les lignes de l'entête ou du pied et l'option `olines` pour les aligner à droite sur la marge extérieure.





# Chapitre 5

## Le jour de la semaine avec `scrdate`

Avec la version 3.05a, les fonctionnalités de ce paq ont été considérablement élargies. En plus de la journée en cours, il peut maintenant déterminer le jour de la semaine d'une date en conformité avec le calendrier grégorien.

```
\CenturyPart{année}  
\DecadePart{année}
```

La commande `\CenturyPart` donne la valeur des chiffres du siècle, centaines et milliers, d'une année tandis que la commande `\DecadePart` donne la valeur des unités et des dizaines restantes. La valeur peut être utilisée directement pour l'affectation à un compteur ou pour des calculs en utilisant `\numexpr`. Pour la sortie en chiffres arabes, faites précéder la commande avec `\the`. Pour calculer et afficher le quantième dans le siècle de l'année en cours :

```
L'année \the\year\ est l'année  
\the\DecadePart{\year} du  
\the\numexpr \CenturyPart{\year}+1\relax  
siècle.
```

Le résultat est :

L'année 2017 est l'année 17 du 21siècle.

Noter que la méthode de comptage utilisée traite l'année 2000 comme étant l'année 0, c'est-à-dire la considère première année du XXI<sup>e</sup> siècle.

<pre>\DayNumber{année}{mois}{jour} \ISODayNumber{date ISO}</pre>
--

Ces deux déclarations renvoient la valeur du numéro de la semaine en retour de n'importe quelle date. Ils ne diffèrent que par la façon d'indiquer la date.

La commande `\DayNumber` indique l'année, le mois et le jour de la date en tant que paramètres séparés, la commande `\ISODayNumber` utilise la date dans la notation ISO, année - mois - jour comme un seul argument. Un ou deux chiffres pour le mois ou le jour n'ont pas d'importance. Le résultat des deux commandes peut être attribué à un compteur ou utilisé par `\numexpr` pour des calculs. Pour la sortie en chiffres arabes, utiliser le préfixe de valeur `\the`.

Pour connaître le numéro du jour de la semaine du 1<sup>er</sup> mai 2027, écrivez donc :

```
le 1\up{er} mai 2027 tombe dans la semaine numéro\dots
\the\ISODayNumber{2027-5-1}.
```

Le résultat :

le 1<sup>er</sup> mai 2027 tombe dans la semaine numéro 6.

Il est même possible d'aller à une date donnée à un jour futur ou passé.

Si vous voulez connaître le numéro des jours de la semaine que nous avons en 12 jours et que nous aurons 24 jours avant le 24 décembre 2027 :

```
Dans 12~jours, nous avons le nombre de jours
\the\DayNumber{\year}{\month}{\day+12} et
24~jours avant le 24~Décembre~2027, c'est le numéro
\the\ISODayNumber{2027-12-24-24}.
```

Il en résulte :

Dans 12 jours, nous avons le nombre de jours de la semaine 1 et 24 jours avant le 24 Décembre 2027, c'est le numéro 2.

Les jours de la semaine sont ici numérotés comme suit : dimanche = 0, lundi = 1, mardi = 2, mercredi = 3, jeudi = 4, vendredi = 5 et samedi = 6.

<pre>\DayNameByNumber{numéro de la semaine} \DayName{année}{mois}{jour} \ISODayName{date ISO}</pre>
---

Habituellement, vous êtes moins intéressé par le numéro du jour de la semaine que par le nom de la journée. Par conséquent, la commande `\DayNameByNumber` renvoie le nom de la journée correspondant à un nombre. Le numéro peut être le

---

résultat qui a été déterminé, par exemple, par `\DayNumber` ou `\ISODayNumber`. Les deux commandes `\DayName` et `\ISODayName` donnent directement le nom du jour de la semaine pour une date donnée. Pour connaître quel jour de la semaine sera le 24 Décembre 2027, écrivez :

```
Compter la quantité de \dots jusqu'à \ISODayName{24/12/2027},
le 24 \12. 2027.
```

Le résultat :

```
Compter la quantité de \dots jusqu'à vendredi 24 12 2027,
```

Précision : une fonction spéciale permet de faire quelques calculs à l'intérieur de l'argument de `\DayName`.

Exemple : vous voulez connaître le nom du jour de la semaine dans 12 jours et celui du jour qui tombe 24 jours avant le 24 Décembre 2027 :

```
Dans 12 jours, ce sera un \DayName{\year}{\month}{\day+12}
et 24 jours avant le 24 Décembre 2027, est \ISODayName
{2027-12-24-24} et trois jours après un mercredi suit le \DayName
ByNumber{3+2*7+3}.
```

Il en résulte :

```
Dans 12 jours, ce sera un jeudi et 24 jours avant le 24 Décembre
2027, est mardi et trois jours après un mercredi suit le samedi.
```

<pre>\ISOToday \IsoToday \todayname \todaynumber</pre>
--

Dans les exemples précédents, la date du jour est toujours déterminée en utilisant explicitement le registre T<sub>E</sub>X `\year`(année) `\month`(mois) `\day`(jour). `\ISOToday` et `\IsoToday` indiquent directement la date du jour dans la notation ISO, et ne diffèrent que par `ISOToday` qui utilise toujours des sorties à deux chiffres, tandis que `\IsoToday` a des sorties de valeurs maximales à 10 chiffres. La commande `\todayname` fournit directement le jour courant, tandis que `\todaynumber` fournit la valeur actuelle du jour.

Plus de détails sur l'utilisation de ces valeurs peuvent être trouvés dans les descriptions précédentes de `\DayNumber` et `\ISODayNumber`. Si je veux vous montrer quel jour de la semaine ce document a été créé, j'écris :

```
Ce document a été créé un \todayname.
```

## 5 Le jour de la semaine avec *scrdate*

---

Le résultat est le suivant : `Ce document a été créé un samedi.`

Noter que le paq n'est pas en mesure de refuser des mots.

Le terme utilisé est reconnu au nominatif singulier, pour la date d'une lettre par exemple. Compte tenu de cette limitation, l'exemple ci-dessus peut ne pas fonctionner correctement dans certaines langues.

Les noms des jours de la semaine sont sauvegardés sous forme de capitales, par exemple la première lettre est une lettre majuscule et les autres sont en minuscules. Mais pour certaines langues, vous devrez peut-être mettre les noms en minuscules, en utilisant la commande L<sup>A</sup>T<sub>E</sub>X `\MakeLowercase` standard :

```
\MakeLowercase\todayname
```

qui permet de convertir son argument entièrement en minuscules. Bien sûr, cette astuce fonctionne également pour les déclarations `\DayName`, `\ISODayName` et `\DayNameByNumber`.

`\nameday{nom}`

Analogue à la façon dont la sortie `\today` peut être modifiée directement à l'aide de `\nameday`, de sorte que le nom de la sortie de `\todayname` peut être changé.

Exemple : vous voulez, avec `\date`, définir une valeur fixe pour la date actuelle. Pour la sortie du jour correspondant de la semaine, il est intéressant de savoir que cette journée était seulement une journée de travail.

Par conséquent, vous écrivez

```
\nameday{journée de travail}
```

et vous obtenez la phrase de l'exemple précédent `\todayname` :

`Ce document a été créé sur une journée de travail.`

Il n'existe pas d'instructions pour `\ISOToday` et `\IsoToday`.

`\newdaylanguage{langue}`

Le paq de *scrdate* connaît et utilise actuellement les langues suivantes :

- allemand (`german`, `ngerman`, `swissgerman`, `nswissgerman`),
- autrichien (`austrian`, `naustrian`),
- anglais (`american`, `australian`, `british`, `canadian`, `english`),
- anglais spécifique (`UKenglish`, `USenglish`),
- croate (`croatian`),
- danois (`danish`),
- espagnol (`spanish`),

- 
- finnois (finnish),
  - français (acadian, canadian, french),
  - italien (italian),
  - néerlandais (dutch),
  - norvégien (norsk),
  - polonais (polish),
  - slovaque (slovak),
  - suédois (swedish),
  - tchèque (czech),

mais peut également être configuré pour d'autres langues. Pour ce faire, on ajoute le premier argument de `\newdaylanguage`, le nom de la langue et d'autres paramètres tels que le nom des jours de la semaine.

Dans la version actuelle, il importe peu de charger `scrdate` avant ou après `babel`, car, dans les deux cas, la langue correcte sera utilisée. Plus précisément : tant que vous utilisez une langue sélectionnée avec `babel`, dans notre cas, `french` ou des paqs similaires compatibles, la langue correcte est reconnue par `scrdate`. Ce qui n'est pas toujours le cas lorsque les termes anglais (US-) sont utilisés.



# Chapitre 6

## L'heure actuelle avec `scrtime`

Ce forfait permet de donner l'heure actuelle. Depuis la version 3.05, il supporte également les classes et, avec option, divers autres paqs de KOMA-Script. Voir par exemple la [section 2.4](#).

```
\thistime[séparation]
\thistime*[séparation]
```

`\thistime` renvoie l'heure actuelle en heures et minutes. L'argument optionnel de séparation, dans la sortie, est réglé entre les heures et les minutes. La valeur est le caractère « : » par défaut.

`\thistime*` fonctionne de la même manière que `\thistime`. La différence est que, avec `\thistime*` le détail des minutes de valeur inférieure à 10 n'est pas précédé d'un zéro pour être étendu à deux chiffres.

Exemple : le train de cette ligne part à `\thistime` précises.

renvoie, par exemple, le résultat :

Le train de cette ligne part à 15:08 précises.

ou :

Le départ de votre train est à 23:09 précises.

En revanche, la ligne fournit :

au prochain top, il sera `\thistime*[\heure,\]` minutes et secondes avec pour résultat possible quelque chose comme :

au prochain top, il sera 8 heures, 41 minutes et 5 secondes.

ou :

au prochain top, il sera 23 heures, 9 minutes et 5 secondes.

`\settime{valeur}`

`\settime` met les sorties `\thistime` et `\thistime*` à une valeur fixe. L'argument de `\thistime` ou `\thistime*` est, par la suite, ignoré, car la chaîne complète `\thistime` ou `\thistime*`, fournie maintenant, a été explicitement définie par les présentes.

`12h=on, off, value(simple switch)`

L'option `12h` peut être choisie, si le temps avec `\thistime` et `\thistime*` devait être publié au format 12 heures ou 24 heures. Une valeur ponctuelle peut être utilisée par défaut pour de simples interrupteurs (voir la table 1.6 page 20). Si la valeur de l'option sans spécification est utilisée, la valeur réelle est supposée, c'est-à-dire que le format 12 heures est retenu. Par défaut, ce sera le format de 24 heures.

L'option peut être soit une option de classe dans `\documentclass`, soit une option de paq dans `\usepackage` ou être réglée après le chargement de `scrtime` par `\KOMAOPTIONS` ou `\KOMAOPTION` (voir par exemple la section 2.4).

Toutefois, vous perdez le contrôle `\settime` de la validité. L'heure affichée utilise cette déclaration seulement ici avec la valeur indiquée dans le format d'émission utilisé. Il est possible avec `\documentclass` et `\usepackage`, mais uniquement pour des raisons de compatibilité avec les versions antérieures de `scrtime`, de passer à l'option `24h` du format 24 heures, mais cette utilisation, considérée comme obsolète, n'est pas recommandée.



# Accès aux fichiers adresses scraddr

## 7.1 Vue d'ensemble

Le paq `scraddr` est un petit plus de la classe lettre de KOMA-Script. Le but est de simplifier l'utilisation des fichiers d'adresses et de rendre leur application plus souple. Fondamentalement, le paq implémente un mécanisme de chargement des fichiers d'adresses, `\adrentry` et `\addrentry`, comme décrit au chapitre 4.

```
\InputAddressFile{nom de fichier}
```

La commande `\InputAddressFile` est la commande principale de `scraddr`. Elle lit le contenu du fichier donné comme argument. Si le fichier d'adresses n'existe pas elle retourne un message d'erreur et pour chaque entrée, elle génère un ensemble de macros pour accéder aux données ce qui, pour les grands fichiers d'adresses, va prendre beaucoup de mémoire T<sub>E</sub>X.

```
\adrentry{nom}{prénom}{adresse}{tel}{F1}{F2} {commentaire}{abrégé}
\addrentry{nom}{prénom}{adresse}{tel}{F1}{F2} {F3}{F4}{abrégé}
\adrchar{initiale}
\addrchar{initiale}
```

La structure des entrées d'adresses dans le fichier d'adresses a été examinée en détail à la section 4.22. La division du fichier d'adresses, également mentionnée en utilisant `\adrchar` ou `\addrchar`, n'a aucun sens pour `scraddr` et elle est

ignorée par le paq. Les commandes d'accès aux données correspondant aux noms des arguments sont en anglais.

```
\Name{abrégé}  
\FirstName{abrégé}  
\LastName{abrégé}  
\Address{abrégé}  
\Telephone{abrégé}  
\FreeI{abrégé}  
\FreeII{abrégé}  
\Comment{abrégé}  
\FreeIII{abrégé}  
\FreeIV{abrégé}
```

L'accès sera basé sur l'abrégé dans le dernier argument d'une entrée, c'est-à-dire l'argument numéro huit pour les entrées de `\adrentry` ou l'argument numéro neuf dans `\addrentry`. Cela signifie également que cet argument ne peut pas être vide. Pour garantir un fonctionnement en toute sécurité, il est recommandé de créer l'abréviation uniquement avec une suite de lettres sans utiliser de trémas.

Si le fichier contient plus d'une entrée avec la même valeur de clé, la dernière occurrence sera utilisée.

## 7.2 Utilisation

Tout d'abord, nous avons besoin d'un fichier d'adresses avec des entrées d'adresses valides. Dans cet exemple, le fichier a pour nom `lotr.adr` et contient les rubriques suivantes :

```
\addrentry{Brutin}{Paulo}  
{La renardière  
\Dans la côte des bourguignons}{}  
{Bilbo Brutin}{fumeur de pipe}  
{porteur d'anneau}{héritier de bilbo}{PAULO}  
\adrentry{Gaudiche}{Sam}  
{3 chemin de la butte  
\La Plaine}{}  
{Rosie Coton}{chiffonnier}  
{porteur de l'anneau du compagnon fidèle}{SAM}
```

```

\adrentry{Bombadil}{Tom}
{la vieille forêt}{}
{Goldbeere}{gazouillis de chansons absurdes}
{Maître de la forêt, de l'eau et de la montagne}{TOM}

```

Le quatrième argument, numéro de téléphone, a été laissé en blanc. Tout d'abord, il n'a aucune importance dans le contexte, et d'autre part, cela devrait en effet être possible. La commande `\InputAddressFile` permet de charger le fichier d'adresse dans notre document lettre :

```
\InputAddressFile{lotr}
```

À l'aide des commandes présentées dans ce chapitre, nous pouvons maintenant écrire une lettre au vieux Tom Bombadil, pour lui demander s'il se rappelle de deux compagnons de voyage des jours anciens :

```

\begin{letter}{\Name{TOM}}
\Address{TOM}
\opening{Cher \FirstName{TOM} \LastName{TOM},}
{FreeIII{TOM}, mon cher {FreeI{TOM},
Te rappelles-tu d'un certain M. \LastName{PAULO}, plus pré-
cisément, \Name{PAULO}, le seigneur \FreeI{PAULO} dans mon
souvenir. Il était \Comment{PAULO} dans son troisième âge et
\FreeIV{PAULO} toujours accompagné par \Name{SAM}, \Comment
{SAM}.
Leur passion était très mondaine \FirstName{PAULO} adorait
fumer et \FreeII{PAULO} appréciait un bon repas avec son ami.
\FreeII{SAM} Vous souvenez-vous?
Mithrandir vous a certainement beaucoup parlé d'eux, de leurs
actions et de leurs aventures.
\closing{
O printemps et l'été et à nouveau le printemps! \
O vent sur la cascade, et les rires des feuilles !
\end{letter}

```

Utilisée dans `\FirstName{abrégé}` et `\LastName{abrégé}`, la composition de cet exemple<sup>1</sup> peut également être obtenue directement au moyen de `\Name{abrégé}`. Dans l'adresse de lettres, tant le nom que le prénom sont souvent obligatoires,

---

1. version originale du poème.

*O Frühling und Sommerzeit und danach wieder Frühling!*

*O Wind auf dem Wasserfall und Lachen des Laubes!*

comme indiqués ci-dessus dans `\opening`. La commande `\Name{clé}` est un abrégé pour `\FirstName{clé} \LastName{clé}`.

Les cinquième et sixième arguments de `\adrentry` et `\addrentry` sont libres. Ils sont accessibles avec les commandes `\FreeI` et `\FreeII`. Dans cet exemple, le cinquième argument contient le nom d'une personne importante dans la vie de la personne de l'entrée et le sixième contient la passion de cette personne. Le septième paramètre est un commentaire ou peut être aussi un paramètre libre. L'accès se fait par `\Comment` ou `\FreeIII`. L'utilisation de `\FreeIV` n'est valable que pour l'entrée `\addrentry`. Dans l'entrée `\adrentry`, son utilisation n'est pas autorisée et provoque une erreur.

### 7.3 Paq avec options d'avertissement

Comme mentionné dans la section précédente, l'utilisation de la commande `\FreeIV` pour `\adrentry` n'est pas autorisée. La réaction de *scraddr* dans une telle situation peut être configurée par les options de *paq*.

```
adrFreeIVempty
adrFreeIVshow
adrFreeIVwarn
adrFreeIVstop
```

TABLE 7.1 – Quatre options permettant à l'utilisateur de choisir parmi quatre réactions différentes entre ignorer et annuler pendant l'exécution de  $\LaTeX$  si la commande d'accès `\FreeIV` a été utilisée avec une entrée `\adrentry`

<code>adrFreeIVempty</code>	la commande <code>\FreeIV</code> est ignorée.
<code>adrFreeIVshow</code>	avertissement : « entry FreeIV undefined@key » écrit dans le texte.
<code>adrFreeIVwarn</code>	écrit un avertissement dans le fichier journal.
<code>adrFreeIVstop</code>	l'exécution de $\LaTeX$ est interrompue avec un message d'erreur.

Utilise le préreglage `adrFreeIVshow` si aucune option n'est définie.

# Chapitre 8

## Création de fichiers d'adresses

Dans les anciennes versions de KOMA-Script, le paq `addrconv` était partie intégrante du système KOMA-Script. Les principaux liens font que, à l'aide de `addrconv` il est possible de créer, à partir d'une base de données d'adresses au format `\bib`, des fichiers compatibles avec la classe lettre KOMA-Script ou le paq `scraddr` :

```
@address {HMUS,
name = {Jean Bambois},
title = {Mag. art.},
city = {La Bambouseraie},
zip = 30140,
country = {France},
street = {rue de Montsauve 552},
phone = {04 66 61 94 94},
note = {seulement toutes les inventions},
key = {HMUS},
}
```

Avec des entrées telles que celles données ci-dessus, les fichiers d'adresses peuvent être générés. Pour ce faire, `addrconv` emploie `\bib` et différents styles de `\bib`. En outre, des fichiers spéciaux de L<sup>A</sup>T<sub>E</sub>X permettent de créer différentes listes d'adresses téléphoniques et de répertoires pour l'impression.

Le paq `addrconv` était un paq indépendant qui comportait, en plus du nécessaire, plusieurs caractéristiques des plus intéressantes.

Attention : il a, depuis un certain temps, été remplacé par le paq `adrconv`, avec un seul « d ».

Si ce dernier n'est pas inclus dans votre distribution `TEX`, il peut être téléchargé à partir de [**Kie10**] et installé séparément.

## Développer les entités `scrextend`

Il existe plusieurs caractéristiques qui sont partagées par toutes les classes KOMA-Script. Cela signifie non seulement les classes `scrbook`, `scrreprt` et `scrartcl` qui ont été créées en remplacement des classes standard `book`, `report` et `artcl` abandonnées, mais aussi pour plusieurs caractéristiques de la classe `scrletter2` de KOMA-Script, qui est conçu comme le successeur de `scrletter` pour les lettres. Ces options de base, qui peuvent être trouvées dans les catégories mentionnées ci-dessus, sont fournies « prêtes à l'emploi » par le paq `scrextend` depuis KOMA-Script 3.00. Ce paq ne doit pas être utilisé conjointement avec une classe KOMA-Script, mais peut être utilisé avec de nombreux autres classes. La tentative de charger ce paq avec une classe KOMA-Script, est reconnue par `scrextend` et rejetée avec un message d'avertissement.

Il n'existe aucune garantie que `scrextend` fonctionne avec toutes les classes. Il est principalement réservé à l'expansion de la classe `standard` et des classes dérivées. Dans tous les cas, les utilisateurs doivent d'abord vérifier si la classe utilisée offre, pas elle-même, des possibilités appropriées.

En plus des options décrites dans ce chapitre, des caractéristiques communes supplémentaires existent, fournies principalement pour les auteurs de classes et de paqs. Celles-ci peuvent être trouvées dans le chapitre 11. Ainsi l'ensemble `scrbase` décrit à ce chapitre a été conçu pour être utilisé principalement par les auteurs de classes et de paqs. Le paq `scrextend` et toutes les classes KOMA-Script utilisent aussi ce paq.

Les classes KOMA-Script et le paq `scrextend` chargent également le paq `scrfile` décrit dans le chapitre 12. Pour cette raison les caractéristiques de ce paq sont également disponibles lors de l'utilisation de `scrextend`.

À la différence de ce qui précède, seules les classes KOMA-Script `scrbook`, `scrreprt`, et le paq `tocbasic` de `scrartcl` (voir le chapitre 14), ont été conçus pour être utilisés par les auteurs de classes et de paq. Les possibilités de `tocbasic` se trouvent donc définies dans ces classes et non pas dans `scrextend` qui ne fournit pas les caractéristiques de ce paq. Néanmoins, vous pouvez utiliser `tocbasic` avec `scrextend`.

## 9.1 Option de sélection précoce ou tardive

Ce qui a été écrit dans la section 2.4 est généralement applicable *mutatis mutandis* (une fois les changements nécessaires effectués). Donc, si vous avez déjà lu et compris la section 2.4, vous pouvez continuer à la section 9.3.

Cette section décrit une caractéristique du programme présenté, concernant non seulement les classes et `scrextend` mais aussi d'autres paqs KOMA-Script. Ainsi, l'utilisateur trouvera dans cette section toutes les informations sur un paq ou une classe. Ces dernières figurent en une forme presque identique dans plusieurs chapitres. Si vous n'êtes pas spécialement intéressés par un guide sur un paq ou une classe, mais souhaitez simplement obtenir une vue d'ensemble de KOMA-Script, vous pouvez lire cette section dans l'un des chapitres, puis l'ignorer lors d'une étude plus approfondie du manuel.

```
\documentclass[liste des options]{classe KOMA-Script}
\usepackage[liste des options]{liste des paqs}
```

Dans L<sup>A</sup>T<sub>E</sub>X est prévu que les utilisateurs spécifient les options de classes sous la forme d'une simple liste de mots-clés séparés par des virgules comme argument optionnel de `\documentclass`, sauf que, dans une classe, elles sont transmises également à tous les paqs qui les comprennent. La procédure est la même avec `\usepackage`. KOMA-Script étend le mécanisme des options pour les classes KOMA-Script (et quelques paqs) à d'autres possibilités en ajoutant une valeur supplémentaire. Ainsi, une option n'est pas limitée à sa forme mais peut aussi prendre la forme `option=valeur`. pour être utilisée avec `\documentclass` et `\usepackage` comme il est décrit dans [Tea05b] ou toute introduction de L<sup>A</sup>T<sub>E</sub>X, par exemple [DGS+12].



Les options de réglage `\documentclass` et `\usepackage` ont un inconvénient qui doit être observé : certains paramètres de longueurs, compteurs, etc. peuvent facilement casser et doivent être attentivement surveillés. Ainsi, l'utilisation d'une option de longueur de  $\text{\LaTeX}$  provoque un message d'erreur avant que la valeur de `KOMA-Script` ne soit adoptée. L'affectation de valeurs avec des longueurs ou des compteurs de  $\text{\LaTeX}$  ne doit jamais précéder `\documentclass` ou `\usepackage`, mais toujours suivre les instructions documentées `\KOMAoptions` ou `\KOMAoption`.

```
\KOMAoptions{titliste d'options}  
\KOMAoption{option}{liste des valeurs}
```

`KOMA-Script` offre la possibilité de modifier la valeur des options de la plupart des classes et des paqs. Certaines options ont aussi une valeur par défaut qui est automatiquement appliquée si la valeur de l'option est omise.

Certaines options peuvent contenir simultanément plusieurs valeurs. Il est alors possible de leur assigner avec `\KOMAoption` une option séquentielle d'un ensemble de valeurs. Les valeurs individuelles sont, dans la liste de valeurs, séparées les unes des autres par des virgules. Si une valeur contient un signe égal ou une virgule, alors la valeur est mis entre accolades.

Pour la réalisation de cette possibilité de commandes `KOMA-Script` utilise `\FamilyOptions` et `\FamilyOption` avec la famille « `KOMA` », voir dans la Partie II, la section 11.3.

## 9.2 Compatibilité avec les versions précédentes

Ce qui a été écrit dans la section 2.5 est applicable *mutatis mutandis* (une fois que les changements nécessaires ont été effectués). Donc, si vous avez déjà lu et compris la section 2.5, vous pouvez passer à la section 9.4.

Archiver les documents dans le code source est généralement d'une importance capitale pour toujours obtenir exactement le même résultat dans les futures cycles d'exécution de  $\text{\LaTeX}$  car, dans certains cas, des améliorations ou des corrections apportés à une classe mènent à des changements de comportement, en particulier lors de la mise en page.

<pre>version=wert version=first version=last</pre>
--

Dans `scrextend` le choix est de savoir si un fichier source devrait, dans la mesure du possible, conduire  $\LaTeX$  exactement au même résultat, ou si les ajustements respectivement fixés avec la dernière version doivent produire une compatibilité de version. Les plus anciennes prises en charge par la version `KOMA-Script` peuvent tout d'abord avec la `version=first`, la `version=2.9` ou la `version=2.9t` être impactées. Lorsque vous spécifiez en tant que valeur une version inconnue, un avertissement est émis et `version=first` est adopté par précaution.

Avec `version=last` la dernière version peut être sélectionnée et dans ce cas, une compatibilité future ommise. Si l'option est utilisée sans valeur, il est supposé que `version=last` est également la valeur par défaut, autant qu'aucune option obsolète n'est utilisée.

Vous utilisez une option obsolète ? `KOMA-Script3` définit automatiquement une version avec `version=first`. Un avertissement explique comment éviter ce commutateur de compatibilité. Alternativement, vous pouvez sélectionner une version différente de l'option de réglage même après l'option obsolète elle-même. La question de la compatibilité concerne principalement la mise en page. De nouvelles options qui ne touchent pas à cette mise en page sont disponibles même si vous avez sélectionné l'option en utilisant la compatibilité avec une version plus ancienne. Elle n'a pas d'effet sur les changements qui résultent de l'élimination des erreurs manifestes lors de l'utilisation d'une version plus récente.

Si vous avez besoin d'une compatibilité de mise en page par défaut, archiver votre document avec la version `KOMA-Script` utilisée. Il est important de noter que si la version `scrextend` est chargée après le `paq`, l'option ne peut pas être changée. Le réglage avec `\KOMAOPTIONS` ou `\KOMAOPTION` lui est destiné.

---

TABLE 9.1 – Aperçu des fonctionnalités étendues en option de `scrextend`

---

`titel`

Les options des classes `KOMA-Script` sont étendues aux pages de titre. Cela concerne non seulement les instructions relatives à la couverture mais aussi les commandes pour la page de titre (voir l'article 10.7)

---

## 9.3 Des fonctions étendues en option

Le paq `scrextend` fournit, en option, des fonctionnalités étendues. Elles ne sont pas disponibles par défaut, mais peuvent être activées en plus, bien que souvent en conflit avec les possibilités des classes `standard` des paqs utilisés.

`extendedfeature=possibilité`

Avec cette option, une fonction étendue de `scrextend` peut être activée.

Disponible uniquement pendant le chargement du paq `scrextend`, elle est installée comme un argument optionnel de `\usepackage{scrextend}`. Une vue d'ensemble de toutes les fonctions disponibles en option est présentée dans la table 9.1 page ci-contre.

## 9.4 Développer en mode création

Ce qui est écrit dans la section 2.3 s'applique *mutatis mutandis* (une fois que les changements nécessaires ont été effectués). Donc, si vous avez déjà lu et compris la section, vous pouvez passer à l'article 9.5 après la fin de la présente section.

De nombreuses classes et de nombreux paqs utilisent le mode de composition normal ainsi que le mode brouillon. Les différences entre ces deux sont aussi diverses que les classes et les forfaits qui offrent cette distinction.

`draft=on-of-valeur`

Cette option distingue les documents du projet des documents finaux lorsque la valeur de marche-arrêt peut être l'une des valeurs standard pour les interrupteurs simples de la table 2.5 page 59. Lorsqu'elle est activée, dans le cas de lignes excessivement longues, une petite boîte noire est émise à la fin de la ligne. Les boîtes aident l'œil non averti pour trouver les paragraphes à traiter manuellement. En revanche, cette boîte n'apparaît pas dans le projet `défait=false`. Ces lignes disparaissent d'ailleurs souvent lors de l'utilisation du paq `microtype` [SCH13].

## 9.5 Sélection de la taille de police du document

Ce qui est écrit dans la section 3.5 s'applique *mutatis mutandis* (une fois que les changements nécessaires ont été effectués). Donc, si vous avez déjà lu et

compris la `section 3.5`, vous pouvez continuer à la fin de cette `section` chapitre 9.6. La police de base et la taille sont des éléments clés dans la conception d'un document. Comme indiqué dans le chapitre 2, la division entre l'espace d'impression et les marges en dépendent en grande partie. La fonte de base est celle utilisée pour la majeure partie du texte d'un document. Tous les réglages dépendants, que ce soit la forme, l'épaisseur, l'inclinaison ou la taille, sont liés à la police de base.

```
fontsize=taille
```

Bien que seul un nombre très limité de tailles de polices est pris en charge par les classes `standard` et la plupart des autres classes, `KOMA-Script` offre la possibilité de spécifier toutes les tailles pour la police de base qui peuvent être utilisées comme unité de taille `TEX` connue. Lorsque la taille est spécifiée sans unité, le `pt` est supposé être l'unité par défaut.

Si l'option est définie dans le document, la taille de la police de base et toutes les variables dépendantes sont modifiées. Cela peut par exemple être utile si l'annexe doit être placée dans une plus petite taille de police. Il convient de noter que, après le chargement la division de classe entre l'espace d'impression et les marges ne sera pas recalculé automatiquement (voir `\recalctypearea`, `section 2.6`). Si ce recalcul est fait, il est basé sur la validité de la taille de la police de base. L'impact de la modification de la taille de police de base dépend des paqs chargés ou de la classe utilisée. Il peut donc se produire des erreurs qui ne sont pas considérées comme des erreurs de `KOMA-Script`.

Cela ne devrait en aucun cas être interprété comme un remplacement pour `\fontsize` (voir [Tea05a]) et ne devrait pas être utilisé à la place de la police de base, les instructions s'appliquant aux polices utilisées de taille `tiny` à `Huge`.

## 9.6 Étiquetage du texte

Ce qui est décrit à la `section 3.6` s'applique *mutatis mutandis*. Les noms et les significations des éléments individuels sont listés dans la table 3.2 page 176. Toutefois, seuls les éléments énumérés pour le titre du document, notes de pied de page, énonciation intelligente et environnement en matière d'étiquetage, sont pris en charge.

Une bonne disposition des éléments existe, elle sera également utilisée pour le titre du document uniquement. Cela a été fait pour assurer la compatibilité

avec les classes KOMA-Script. Les valeurs par défaut sont indiquées dans les paragraphes correspondants.

```
\textsuperscript{texte}
\textsubscript{texte}
```

Dans la déclaration du noyau L<sup>A</sup>T<sub>E</sub>X `\textsuperscript` est déjà défini et permet de placer le texte en indice depuis la version 1.1.2015. Pour les anciennes versions L<sup>A</sup>T<sub>E</sub>X KOMA-Script définit `\textsubscript`. Un exemple d'application peut être trouvé dans la section 3.6.

```
\setkomafont{élément}{commande}
\addtokomafont{élément}{titcommande}
\usekomafont{commande}
```

En suivant les instructions `\setkomafont` et `\addtokomafont` il est possible de définir les commandes qui modifient les caractéristiques d'un élément particulier commuté. Théoriquement du texte littéral (comme des commandes) pourrait être utilisé dans toutes les directions possibles. Cependant, vous devez nécessairement limiter ces déclarations qui n'ont vraiment qu'un seul attribut de police commuté. Habituellement, ces commandes sont comme `\rmfamily`, `\sffamily`, `\ttfamily`, `\mdseries`, `\bfseries`, `\normalfont`, `\upshape`, `\itshape` et puis `\slshape`, `\scshape` ou l'une des commandes `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize`, `\small`, `\footnotesize`, `\scriptsize` et `\tiny`. Pour l'explication de ces commandes, consulter [DGS+12], [Tea05a] ou [Tea05b]. Les commutations de couleur sont possibles avec `\normalcolor` (voir [Car05] et [Ker07]). Les autres commandes qui permettent la redéfinition, ou gèrent l'édition, ne sont pas fournies. Un comportement étrange est possible mais il ne constitue pas un bug.

L'élément de commutation de la police, `\setkomafont` fournit une définition nouvelle. En revanche, l'existante est seulement étendue avec `\addtokomafont`. Il est recommandé d'utiliser les deux directives dans le même document, mais seulement dans le préambule du document. Pour des exemples de l'utilisation, se référer aux sections sur les différents éléments, les noms et significations des éléments individuels sont énumérés dans la table 3.2 page 176.

Dans `scrextend`, seuls les éléments du titre du document, maxime, notes et environnement d'étiquetage sont traités, bien que l'élément disposition utilisé par `scrextend` seulement pour le titre du document est également disponible. Les paramètres par défaut sont décrits dans les sections respectives. Avec la déclaration `\usekomafont` la police en cours peut être changée avec celle définie pour l'élément spécifié.

Si vous souhaitez mettre le titre en **serif** rouge, ce peut être fait facilement avec :

```
\setkomafont{title}{%  
\color{red}%  
}
```

Pour la commande `\color{red}` le paq `color` ou `xcolor` est nécessaire. Dans cet exemple, l'indication supplémentaire `\normalfont` ne l'est pas parce que cette déclaration est déjà contenue dans la définition du titre. L'exemple suppose que `extendedfeature=titre` est défini (voir [section 10.3](#)),

```
\usefontofkomafont{élément}  
\useencodingofkomafont{élément}  
\usesizeofkomafont{élément}  
\usefamilyofkomafont{élément}  
\useseriesofkomafont{élément}  
\useshapeofkomafont{élément}
```

Parfois, lors de l'écriture principale d'un élément, des choses sont faites qui n'ont rien à voir avec l'écriture. Même si cela est expressément déconseillé, régler simplement la police et les paramètres additionnels en utilisant `\usekomafont` et `\usefontofkomafont`.

Cette déclaration accepte les tailles de la police et de la distance de base, le codage (eng. encoding), la famille (eng. family), l'épaisseur du trait ou la mise en forme (eng. fontseries) ainsi que la forme et l'alignement (eng. fontshape).

Les instructions restantes ainsi que les attributs individuels de police sont acceptés ici par `\usesizeofkomafont` qui gère à la fois les tailles de la police et de la distance de base.

Ces commandes ne doivent pas être interprétées comme une légitimation pour toutes les instructions d'insertion dans les paramètres de la police des éléments. Cela peut très rapidement conduire à des erreurs (voir [section 21.3](#)).

## 9.7 Titres de documents

Ce qui est décrit dans la [section 3.7](#) s'applique *mutatis mutandis*. Mais il existe une différence : les capacités de titre de document de `scrextend` font partie des options, des fonctionnalités avancées, donc ne sont disponibles que si `extendedfeature=title` a été sélectionné lors du chargement du paq

(voir la section 10.3). Et `scrextend` ne peut pas être utilisé avec une classe KOMA-Script. Dans tous les exemples de la section 3.7, pour l'utilisation de `scrextend`

```
\documentclass{scrbook}
doit par conséquent être remplacé par
\documentclass{book}
\usepackage{extendedfeature=title}{scrextend}
```

On distingue deux types de titres. Tout d'abord, il y a la première page avec le titre du document avec des informations supplémentaires tel l'auteur sur une page séparée. En plus de la page de titre principal, il peut comporter plus avant, un demi titre, des informations sur l'éditeur, une dédicace ou similaire. Deuxièmement, il y a le titre dans l'entête qui apparaît seulement au début d'une nouvelle page. En dessous de cette ligne de titre, par exemple avec le résumé de l'introduction, la continuation du texte ou la table des matières.

```
titlepage=on-off-valeur
titlepage=firstiscover
```

Cette option est sélectionnée pour titre de couverture si elle est définie par `\maketitle` et ses propres pages sont utilisées, ou bien le titre de couverture est fixé par `\maketitle` comme titre de l'entête. La valeur de marche-arrêt peut être l'une des valeurs standard pour les interrupteurs simples de la table 2.5 page 59.

Avec `titlepage=true`, le titre de couverture est sélectionné sous la forme de page de couverture. La commande `\maketitle` utilise cet environnement pour la mise en page du titre de cette page qui, normalement, ne reçoit ni entête, ni pied de page. Avec KOMA-Script le titre de couverture est grandement élargi par rapport aux classes `standard`.

Le contraste est réalisé avec `titlepage=false` qui définit le titre de l'entête. Autrement dit, le titre de couverture sera spécialement mis en évidence, sur la page avec le titre, mais peut être suivi d'une autre matière comme un résumé ou une `section`.

Avec la troisième option, `titlepage=firstiscover`, les premières pages ne sont pas les seules activées. Il est également veillé à ce que la première sortie de page avec `\maketitle`, soit le demi-titre ou le titre principal, se fasse comme page de couverture. Tout autre paramètre de l'option titre de page remplace à nouveau ce paramètre. Les bords de cette page de couverture sont déterminés par `\coverpagetopmargin` (bord supérieur), `\coverpageleftmargin` (bord

gauche), `\coverpagerightmargin` (marge droite) et `\coverpagebottommargin` (bord inférieur). Les réglages pour les longueurs dépendent de `\topmargin` et `\evensidemargin` par défaut et peuvent être changés avec `\renewcommand`.

Le réglage par défaut dépend de la classe utilisée, reconnu par `scrextend` et compatible avec les classes `standard`. Si une classe ne comporte aucun pré-réglage correspondant, le titre de l'entête est défini par défaut.

```
\begin{titlepage}...\end{titlepage}
```

Fondamentalement, les classes `standard` et, avec `KOMA-Script` toutes les pages de titre sont fixées dans un environnement spécial, `titlepage` qui commence toujours une nouvelle page - dans la mise en page recto-verso une nouvelle page de droite - en mode une colonne. Pour un côté le style de page est modifié avec `\thispagestyle{empty}`, de sorte que ni le numéro de page, ni le titre en cours d'exécution ne sont émis. À la fin de l'environnement, la page est automatiquement achevée. Si vous ne pouvez pas utiliser la mise en page automatique du titre, `\maketitle` fournit la description ci-dessous d'un environnement recommandé pour créer votre propre conception.

Pour un exemple de page de titre simple, voir la [section 3.7](#).

```
\maketitle[numéro de page]
```

Alors que les classes `standard` produisent une page de titre avec éléments existants : titre, auteur et date, `\maketitle` permet de régler jusqu'à six pages dans `KOMA-Script`. Contrairement aux classes `standard`, `\maketitle` accepte avec `KOMA-Script` un argument numérique optionnel. S'il est utilisé, le nombre sera le numéro de la première page de titre. Ce numéro de page n'est pas édité mais affecte uniquement la numérotation. Vous devez ici nécessairement choisir un nombre impair. À mon avis, il n'existe que deux applications significatives pour l'argument optionnel. D'une part, on pourrait donner à la page du demi-titre le numéro de page logique -1, de façon à démarrer à 1 le numéro de la page de titre principal. D'autre part, on pourrait commencer par un nombre plus élevé de page, tels que 3, 5 ou 7 pour accueillir d'autres pages de titre qui sont ajoutées par l'éditeur. Si un titre d'entête est utilisé l'argument optionnel est ignoré. Le style de page d'un tel titre peut être modifié en redéfinissant la macro `\titlepagestyle` (voir la [section 3.12](#)).

Les commandes suivantes ne conduisent pas à la production de titre qui se fait toujours avec `\maketitle`. Il convient de noter à ce stade que `\maketitle` ne doit pas être utilisé dans un environnement `\titlepage`. Comme il est indiqué dans les exemples, vous ne devez utiliser que `\maketitle` ou `titlepage`.



Avec les commandes décrites ci-dessus, aucun contenu de titre n'est fixé. Vous devez donc nécessairement utiliser avant `\maketitle`. Cependant, il est nécessaire lorsque vous utilisez le paq `babel` (voir [BB13]) et également recommandé de ne pas l'utiliser dans le préambule avant `\begin{document}`. Des exemples peuvent être trouvés à la fin de la section.

```
\extratitle{demi-titre}
```

Auparavant, le bloc de livre a été souvent protégé par une couverture contre la pollution. Cet objet a ensuite pris la première page du livre qui portait essentiellement un titre abrégé, précisément la moitié de titre. Aujourd'hui, la page supplémentaire est souvent appliquée avant le vrai titre complet et contient la publication de l'information, le numéro de série du livre et des informations similaires.

Avec `KOMA-Script`, il est possible d'inclure une page avant la page de titre réel. Le demi-titre peut être réglé d'une façon arbitraire comme un texte, même avec plusieurs paragraphes. Le contenu du demi-titre émis avec `KOMA-Script` n'influence pas sur la mise en forme de sortie qui dépend du bon vouloir de l'utilisateur.

Le dos du demi-titre reste vide, mais il a aussi sa propre page de titre avec des entêtes de titre. La sortie `\extratitle` est définie dans le cadre des titres produits par `\maketitle`.

Un exemple de page d'accueil simple avec demi-titre et titre principal se trouve dans la section 3.7.

```
\titlehead{entête}
\subject{sujet}
\title{titre}
\subtitle{sous-titre}
\author{auteur}
\date{date}
\publishers{éditeur}
\and
\thanks{note}
```

Sept éléments sont disponibles pour le contenu de la page de titre. L'édition de la page de titre principal a lieu dans le cadre des titres produits par `\maketitle` tandis que les instructions fournies ici ne concernent que la définition des éléments correspondants.

L'entête du titre principal est défini avec la déclaration `\titlehead`. Il s'étend sur toute la largeur du texte, justifié dans des conditions normales au début de la page. Il peut être librement conçu par l'utilisateur. Le sujet est sorti juste au-dessus du titre, dans la même police que ce dernier (voir la table 2.4 page 54).

Le titre est affiché dans une très grande police. Un changement de format de la police pour l'élément de titre restera sans effet (voir la table 2.4 page 54).

Le sous-titre est sorti dans la même police juste en dessous du titre (voir la table 2.4 page 54). Le sous-titre est suivi par l'auteur. Il peut être spécifié, dans l'argument `\author`, plus d'un auteur. Les auteurs sont séparés l'un de l'autre par `\and`. La sortie est écrite dans la même police (voir la table 2.4 page 54).

L'auteur ou les auteurs sont suivis de la date. Voici la date actuelle, `\today`, par défaut. On peut également avec `\date` n'entrer aucune indication et obtenir un blanc. La sortie est faite dans la même police (voir la table 2.4 page 54).

Enfin vient l'éditeur. La saisie peut être réalisée sur toute la largeur de la page en utilisant, si nécessaire, un `\parbox` pour que l'information ne soit pas centrée mais justifiée dans l'ensemble du bloc. Elle est alors considérée comme équivalente à l'entête. Toutefois, noter qu'elle est mise au-dessus des notes de pied de page existantes. Bien sûr, cette commande peut également être utilisée pour d'autres détails de moindre importance. La sortie est dans l'écriture du même élément (voir la table 2.4 page 54).

Sur la première page, les notes de pied de page ne sont pas créées avec `\footnote` mais avec l'instruction `\thanks` et sont généralement utilisées pour des commentaires d'auteurs avec des symboles comme marqueurs de notes au lieu de numéros. Il est à noter que `\thanks` doit être utilisé dans l'argument des autres états comme argument auteur de la déclaration `\author`.

Pour l'élément `\footnote` avec le paq `\scrextend`, le réglage de la police devra tenir compte non seulement de l'extension du titre qui doit être activée mais aussi veiller à ce que les notes soient réglées avec ce paq (voir l'introduction de l'article 10.11). Dans le cas contraire, la police de la classe ou une police prédéterminée est utilisée pour les notes.

Pour la question des éléments de titre, la police peut s'appuyer sur les mêmes éléments en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir la section 9.6). Les réglages par défaut sont indiqués dans la table 2.3 page 53.

Depuis l'entête jusqu'aux éventuelles notes de pied de page, toutes les sorties sont centrées horizontalement. La mise en forme des éléments individuels est encore une fois brièvement résumée dans la table 2.4 page 54.

Un exemple avec tous les éléments du programme est proposé, pour la page de titre principal, dans la **section 3.7**.

Une idée fausse commune concerne le rôle de la page de titre complet. Il est souvent supposé de manière erronée qu'elle implique la couverture du livre ou son enveloppe. Par conséquent, il est souvent prévu que la page de titre ne suit pas les exigences d'une mise en page recto-verso, avec d'aussi grandes marges à droite et à gauche. Toutefois, si l'on prend un livre et qu'on l'ouvre, on rencontre très rapidement au moins une page de titre sous la couverture dans le soi-disant corps du livre. Ces couvertures sont scrupuleusement produites par `\maketitle`.

Comme avec le demi-titre, cela concerne aussi la page de titre complet dans le corps du livre et devrait mener à la même mise en page de l'ensemble du document. Cette couverture du livre peut être quelque chose que vous avez créé dans un document distinct. Elle prend souvent une forme très individuelle et il n'y a aucune raison de ne pas prendre un programme graphique ou une publication assistée par ordinateur pour sa création. Un document distinct qui sera finalement publié plus tard sur un autre support, éventuellement en carton, et avec une autre imprimante, doit être utilisé.

Depuis KOMA-Script 3.12 la première page émise avec `\maketitle` peut être éditée comme page de titre mais aussi être formatée comme page de couverture, et dans ce cas, seules les bordures utilisées sont modifiées (voir l'option `titlepage=firstiscover`).

```
\uppertitleback{titre de l'entête de la page verso}
\lowertitleback{titre du pied de la page verso}
```

Dans l'impression recto-verso, le dos de la feuille avec la page de titre est laissé vide dans les classes **standard**. Avec KOMA-Script, le dos de la page de titre peut être utilisé pour de plus amples informations. Une distinction est faite entre deux éléments que l'utilisateur peut librement concevoir : le titre de l'entête et du pied de page de la page verso.

L'entête peut s'étendre au pied et vice versa. Si l'on prend ce manuel à titre d'exemple, le désistement est défini en utilisant `\uppertitleback`.

```
\dedication{dédicace}
```

KOMA-Script fournit une page de dédicaces. Elles sont centrées par défaut, dans une police légèrement plus grande. Le réglage exact de la police pour les éléments peut être vu dans la table 2.3 page 53 et modifié via les commandes `\setkomafont` et `\adatokomafont` (voir **section 9.6**).

Le côté verso est essentiellement vide comme dans la page avec le faux-titre. La page de dédicace est sortie en même temps que le reste des titres produits par `\maketitle` et doit donc être définie avant cette commande. Un exemple d'ensemble du programme KOMA-Script avec toutes les premières pages offertes est proposé dans la section 3.7.

## 9.8 Détection des pages paires et impaires

Ce qui est décrit dans la section 3.11 s'applique *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 3.11, vous pouvez passer à la section 9.9.

Pour les documents recto-verso la distinction est faite entre les côtés gauche et droit. Ici, un côté gauche a toujours un nombre pair et un côté droit toujours un nombre impair.

`\ifthispageoddalorssinon`

Vous voulez préciser dans KOMA-Script si un texte tombe sur une page paire ou impaire, utiliser l'instruction `\ifthispageodd`. Ici, l'argument « alors » est exécuté si l'on est sur une page impaire. Sinon, l'argument « sinon » est utilisé.

Supposons que vous voulez indiquer si un texte tombe sur une page paire ou impaire. Vous pouvez alors, par exemple, en tapant :

Ceci est une page avec un nombre

```
\lstinline{\Ifthispageodd {im} {} pair}.
```

donne la sortie :

Ceci est une page avec un nombre pair.

Noter que dans cet exemple, l'option avec l'argument « sinon » reste vide.

La commande `\ifthispageodd` utilise un mécanisme similaire avec étiquette et référence, et au moins deux exécutions de L<sup>A</sup>T<sub>E</sub>X sont nécessaires après chaque modification. Seulement alors le résultat est correct. Dans la première manche, une heuristique est utilisée pour la décision.

Plus de détails sur le problème de la détection des côtés gauche et droit, et même des numéros de page pair ou impair peuvent être trouvés pour les experts dans l'article 21.1.

## 9.9 Choix d'un style de page pré-défini

L'une des caractéristiques de base d'un document est le style de page. Style de page dans  $\text{\LaTeX}$  signifie essentiellement entête et pied de page. Le paq `scrextend` ne définit pas de style de page, mais il utilise et prévoit la définition des styles de page.

```
\titlepagestyle
```

Sur certaines pages, un style différent est sélectionné automatiquement en utilisant `\thispagestyle` et `scrextend` l'appliquait sur les seules pages de titre, et même seulement avec la fonction `extendedfeature=title` (voir section 10.3). Quel style de page est-ce réellement ? c'est le style macro `\titlepage`. Par défaut, ce style de page est simple. Il est déjà défini dans le noyau de  $\text{\LaTeX}$  et donc doit toujours être disponible.

## 9.10 Pages intercalaires vierges

Ce qui est écrit dans la section 3.13 s'applique *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 3.13, vous pouvez continuer à la section 9.11.

Des pages blanches sont des pages qui restent intentionnellement vide lorsque l'on définit un document dans  $\text{\LaTeX}$  mais définies par défaut pour le style de page en cours de validité. `KOMA-Script` fournit ici diverses extensions.

Des pages blanches se trouvent principalement dans les livres. D'habitude dans un livre, les chapitres commencent sur le côté droit et dans le cas où le chapitre précédent se termine également sur un côté droit, une page vide est insérée du côté gauche. Il y a aussi des pages normalement vierges qui n'existent que dans la mise en page recto-verso.

```
cleardoublepage=style de page
cleardoublepage=current
```

Cette option vous permet de déterminer le style de page blanche, lorsqu'il est requis par les instructions `\cleardoublepage`, `\cleardoubleoddpage` ou `\cleardoublepage` qui est inséré pour créer la page désirée. Tous les styles de page déjà définis sont tous des styles de page possibles (voir la section 9.9 et le chapitre 5). En outre, `\cleardoublepage=current` est actuellement possible. Ce cas correspond à la valeur par défaut de la version `KOMA-Script2.98c` et

provoque la création d'une page blanche avec le style de page en cours lors de l'insertion.

Depuis la version 3.00 des pages blanches sont par défaut en conformité avec les conventions typographiques créées avec le style de page vide si vous n'avez pas défini la compatibilité avec les versions antérieures de KOMA-Script (voir la version de l'option, section 9.2).

Supposons que vous voulez des pages blanches vides, sauf pour la pagination, produites avec la page de style `plain`. Vous pouvez le faire par exemple avec

```
\KOMAOptions{cleardoublepage=plain}
```

Plus de détails sur le style de page `plain` peuvent être trouvés dans la section 3.12.

```
\clearpage
\cleardoublepage
\cleardoublepageusingstyle{style de page}
\cleardoubleemptypage
\cleardoubleplainpage
\cleardoublestandardpage
\cleardoubleoddpage
\cleardoubleoddpageusingstyle{style de page}
\cleardoubleoddemptypage
\cleardoubleoddplainpage
\cleardoubleoddstandardpage
\cleardoubleevenpage
\cleardoubleevenpageusingstyle{style de page}
\cleardoubleevenemptypage
\cleardoubleevenplainpage
\cleardoubleevenstandardpage
```

Dans L<sup>A</sup>T<sub>E</sub>X noyau existe l'instruction `\clearpage` qui garantit que tous les flottants ne sont pas encore sortis et commence une nouvelle page. En outre, la commande `\cleardoublepage` fonctionne comme `\clearpage` mais seulement dans une mise en page recto-verso (voir option `twoside` dans la section 2.6), et lance un nouveau côté droit. Une page blanche est affichée dans le style de page, à gauche, si nécessaire.

Avec KOMA-Script `\cleardoubleoddstandardpage` fonctionne exactement de la manière décrite pour les classes `standard`.

La commande `\cleardoubleoddplainpage` modifie le style `plain` de la page de gauche vide afin de supprimer le titre de l'entête de colonne. De même, la déclaration `\cleardoubleoddeftypage` de page vide utilisée dans le style `empty` supprime les deux titres de colonnes et le numéro de page du côté gauche. La page est maintenant complètement vide. Vous pouvez spécifier votre propre style de page qui sera pris comme argument de la page blanche avec `\cleardoubleoddpagusingstyle`. Dans ce cas, tout style de page défini précédemment (voir le chapitre 5) peut être utilisé.

Parfois, vous ne voulez pas d'un nouveau chapitre qui commence sur le côté de droite, mais à gauche sur une double page. Bien que cela soit à l'encontre de la typographie classique, cette présentation particulière peut être rendue nécessaire par la présentation d'un contenu plus spécifique. Avec `KOMA-Script` la commande `\cleardoublepage` est définie comme équivalente à la commande `\cleardoubleoddstandardpage`, mais avec la différence que la page suivante est du côté gauche. Cela vaut pour la commande `\cleardoubleevenplainpage`, pour `\cleardoubleevenemptypage` et `\cleardoubleevenpagusingstyle`.

Les commandes `\cleardoublestandardpage`, `\cleardoubleemptypage` et la commande `\cleardoubleplainpage` attendent une commande argument de type `\cleardoublepagusingstyle` en utilisant le style correspond au paquet `scrextend` ainsi que la commande `\cleardoublepage` standard correspondante, précédemment déclarée comme instructions pour la transmission vers la prochaine page impaire.

Supposons que vous utilisez une double page pour placer un document avec une image sur le côté gauche, placée dans la taille de la zone de type et que, du côté droit, un nouveau chapitre commence. Si le chapitre précédent se termine sur un côté gauche, une page vierge est à insérer, qui doit être complètement vide, sans entête ni pied de page. Tout d'abord, avec

```
\KOMAOptions{cleardoublepage=empty}
```

Tout d'abord, les pages blanches avec le style de page `empty`, sont réglées sans entête ni pied de page. Ce réglage peut être déjà fait dans le préambule du document. Les options peuvent également être spécifiées comme arguments optionnels de `\documentclass`.

Au point désiré, vous écrivez maintenant :

```
\cleardoubleevenemptypage
\thispagestyle{empty}
\includegraphics[width=\textwidth,%
height=\textheight,%
```

```
keepaspectratio]%  
{bild}
```

```
\chapter{Intitulé de chapitre}
```

La première ligne change le côté gauche de côté et ajoute pour ce faire, si nécessaire, un côté complètement vide à droite. La deuxième ligne garantit que ce côté gauche est également défini avec le style de page vide. De la troisième à la sixième ligne les commandes chargent l'image nommée fichier image et l'amène à la taille désirée sans la déformer. Pour cela, le paquet utilise `graphicx` (voir [Car05]). La dernière ligne commence - à droite - une nouvelle page de chapitre. Les commandes `\cleardoubleoddpaper` et `\cleardoublepage` conduisent toujours la page à la prochaine partie gauche ou droite. Ici, le style de page est choisi en fonction de l'option `cleardoublepage` et une page vierge éventuellement insérée.

## 9.11 Notes de pied de page

Ce qui est écrit dans la section 3.14 s'applique par analogie. Donc, si vous avez déjà lu et compris la section 3.14, vous pouvez continuer à la section 9.12.

La possibilité de gérer les notes de bas de page des classes KOMA-Script est également fournie par `scrextend`. Par défaut, la mise en forme des notes est toutefois laissée à la classe utilisée, ce qui change dès que la commande `\deffootnote` décrite plus avant dans le texte, est utilisée. Les options de réglage du séparateur de notes ne sont pas fournies par `scrextend`.

<code>footnotes=réglage</code>
--------------------------------

De nombreuses classes marquent par défaut les notes dans le texte avec de petits chiffres en exposant. Plusieurs notes consécutives peuvent donner l'impression qu'il ne s'agit pas de notes distinctes, mais d'une seule avec un nombre élevé. Avec `footnotes=multiple`, les notes se succèdent directement au lieu d'être ensemble dans un délimiteur. Le séparateur `footnotes=multiple` est préconfiguré avec une virgule. L'ensemble du mécanisme est compatible avec la mise en œuvre `footmisc` version 5.3d (voir [Fai11]) affectant les deux marqueurs de bas de page qui ont été fixés avec `\footnote`, ainsi que celles qui sont émises directement avec `\footnotemark`.



Il est possible à tout moment, avec `\KOMAOPTIONS` ou `\KOMAOPTION` de revenir à `footnotes=nomultiple` par défaut. Toutefois, en cas de problèmes avec d'autres paqs qui influent sur les notes, l'option ne doit pas être utilisée et les paramètres ne sont pas modifiés dans le document.

Un résumé des valeurs possibles pour le réglage des notes est fourni dans la table 3.11 page 200.

```
\footnote[numero]{texte}
\footnotemark[numero]
\footnotetext[numero]{texte}
\multiplefootnoteseparator
\multfootsep
```

Les notes s'utilisent dans KOMA-Script comme dans les classes `standard` avec la commande `\footnote` ou avec la paire `\footnotemark` et `\footnotetext`. Comme avec les classes `standard`, il est possible qu'un saut de page se produise. Cela arrive généralement lorsque la marque de bas de page correspondante est placée si bas que la page n'a pas d'autre choix que de développer sur la page suivante. Contrairement aux classes `standard` KOMA-Script offre la possibilité de gérer des notes qui se succèdent directement, en détectant automatiquement et en déplaçant l'autre avec un séparateur. Voir les options déjà documentées.

En appelant `\multiplefootnoteseparator`, ce séparateur peut être placé manuellement. Toutefois, l'utilisateur ne devrait pas redéfinir cette déclaration car en plus du séparateur elle contient les commandes de sa mise en forme, par exemple le choix de la taille des caractères et exposants. Le séparateur lui-même est stocké dans la commande `\multfootsep`. Dans le cadre de ce que :

```
\newcommand*{\multfootsep}{,}
```

définit, cela peut être redéfini. Supposons que vous voulez faire deux notes sur le même mot. Dans la première approche, vous écrivez :

```
Wort\footnote{note 1}\footnote{note 2}.
```

En outre, les notes sont numérotées 1 et 2. Étant donné que les deux numéros se suivent, l'impression produite est que le mot ne comporte qu'une seule note avec le numéro 12. Vous pouvez changer cet effet maintenant en utilisant :

```
\KOMAOPTIONS{footnotes=multiple}
```

qui active la détection automatique des groupes de notes. Vous pouvez également utiliser :

```
Wort\footnote{Fußnote 1}%
```

```
\multiplefootnoteseparator
\footnote{Fußnote 2}
```

qui devrait continuer à fonctionner si la détection automatique échoue pour une raison quelconque, ou ne peut pas être utilisée.

Supposons que vous voulez aussi utiliser les numéros suivis non seulement par une virgule, mais par une virgule séparée par une espace, dans ce cas, écrivez :

```
\renewcommand*{\multfootsep}{, \nobreakspace}
```

dans le préambule de votre document. Ici, `\nobreakspace` a été choisi pour remplacer un espace normal, de sorte que le classement de la note de référence peut être fait sans rupture de paragraphe ou saut de page.

```
\footref{référence}
```

Parfois, vous obtenez une note de bas qui devrait être référencée plusieurs fois dans le texte du document. La mauvaise solution serait l'utilisation de `\footnotemark` spécifiant le numéro souhaité. Cette solution vous oblige à connaître ce numéro qui peut changer à tout moment.

Par conséquent `KOMA-Script` offre la possibilité d'utiliser le mécanisme de `\label` avec également référence aux notes. Il donne un code `\label` à la note correspondante et ensuite avec `\footref` réfère cette `footnote` partout dans le texte.

Vous écrivez un texte dans lequel il est nécessaire de mettre une note à chaque occurrence d'un nom de marque pour indiquer que ce nom est « protégé ». En écrivant :

```
La société SplischSplasch \footnote {Ce nom est une marque déposée
dont les droits sont réservés au propriétaire. \label{refnote}} a
en plus SplischPlumps \footref {refnote} qui fournit une version+
améliorée SplischPlatsch+ \footref {refnote}.
```

La marque est ensuite placée trois fois dans la même note, une fois directement avec `\footnote` et deux fois avec `\footref`.

Étant donné que les marques de notes de pied de page peuvent être définies en utilisant le mécanisme `\label`, les changements qui affectent la numérotation de la note nécessitent éventuellement deux compilations de `LATEX` jusqu'à ce que le résultat des marques par `\footref` soit correct.

```

\deffootnote[marque largeur]{entrée}{détail des
entrées}{définition marque}
\deffootnotemark{définition marque}
\thefootnotemark

```

Les classes KOMA-Script balisent les notes un peu différemment des classes `standard`. La marque dans le texte référence la note, comme cela se fait avec les classes `standard`, par un petit nombre placé en exposant. La même mise en forme est reproduite dans la note elle-même, alignée à droite et de même largeur que le texte principal. La première ligne de la note suit directement le numéro.

Les lignes suivantes sont organisées selon la quantité d'entrées. Si l'option paramètre de largeur n'est pas spécifiée, la valeur par défaut est utilisée pour la mise en retrait. Si la note se compose de plusieurs points, la première ligne d'un paragraphe est en outre fournie avec la taille de tiret du paragraphe.

La figure page 163 illustre les différents paramètres. Le réglage par défaut dans les classes KOMA-Script correspond à la définition suivante :

```

\deffootnote[1em]{1.5em}{1em}{%
\textsuperscript{\thefootnotemark}}

```

Les deux index supérieurs et la taille de police plus petite sont obtenus avec `\textsuperscript`. L'expression `\thefootnotemark` retourne la marque de la note sans aucun formatage. Le paq `scrextend`, cependant, laisse dans le réglage les notes de la classe utilisée par défaut. Le chargement du paq seul ne devrait donner lieu à aucun changement dans la mise en forme des notes ou des marques de bas de page. Pour accepter les paramètres par défaut des classes KOMA-Script, il est préférable de faire soi-même la mise en page. À cet effet, la ligne de code ci-dessus peut par exemple être inséré immédiatement après le chargement `\scrextend`. Dans la note, étiquetage compris, se trouve aussi la mention d'application de la police à l'élément sélectionné. Mais la police de l'étiquette peut avec les commandes `\setkomafont` et `\addtokomafont` (voir la section 9.6) être réglée différemment. Voir aussi la table 3.2. Le réglage par défaut n'entraîne pas de changement de la police. Les éléments trouvés dans `\scrextend` ne sont cependant applicables que si les notes sont établies avec ce paq, c'est-à-dire que `\deffootnote` a été utilisé.

La marque de référence dans le texte est définie séparément de la marque avant la note. Cela se fait avec la déclaration `\deffootnotemark`. Le réglage par défaut est :

```

\deffootnotemark{\textsuperscript{\thefootnotemark}}

```

On trouve dans l'application `\footnotereference` les paramètres (voir la table 3.2 page 176) de l'élément. Les marquages dans le texte et dans la note elle-même sont identiques. La police peut être changée avec les instructions `\setkomafont` et `\addtokomafont`, une fois que `\deffootnotemark` est utilisé (voir la section 9.6). Un aménagement souvent demandé est que la note ne soit ni en exposant ni dans une écriture plus petite. Pour cela, il ne faut pas toucher directement le texte, mais légèrement en avant de celui-ci. Cela peut tout d'abord être réalisé comme suit :

```
\deffootnote{1em}{1em}{\thefootnotemark\ }
```

La marque de référence et l'espace suivante sont ainsi alignés à droite dans une boîte de 1em de largeur. Les lignes du texte de la note suivante sont également en retrait par rapport au bord gauche de 1em.

Une autre forme souvent demandée consiste à aligner à gauche les marques dans la note de pied de page. Ceci peut être obtenu avec la définition suivante :

```
\deffootnote{1.5em}{1em}{%
\makebox[1.5em][l]{\thefootnotemark}%
}
```

Toutefois, si les toutes les notes sont définies dans une police différente, par exemple, **sans serif**, vous pouvez facilement en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir la section 10.6) résoudre le problème :

```
\setkomafont{footnote}{\sffamily}
```

```
\setfootnoterule[hauteur]{longueur}
```

Habituellement, un séparateur est placé entre la zone de texte et les notes, mais généralement ne passe pas sur toute la largeur de la zone de type. Avec cette déclaration, la longueur exacte et la hauteur ou l'épaisseur de la ligne peuvent être déterminées. Cette hauteur et la longueur ne sont eux-mêmes évalués en fonction de `\normalsize` lors de la définition de la ligne. Le niveau de paramètre optionnel peut être complètement éliminé et ne sera pas modifié. Si la hauteur de l'argument ou la longueur est vide, leur taille respective ne change pas. Un avertissement est lancé lors du réglage si des valeurs de tailles non plausibles sont utilisées.

La couleur de la ligne est fixée avec l'élément `footnoterule` en utilisant les instructions `\setkomafont` et `\addtokomafont` (voir la section 9.6). Pas de changement de police ou de couleur par défaut, et le paq `xcolor` doit être chargé pour changer la couleur.

## 9.12 Citations et maximes

Ce qui est décrit [section 3.17](#) s'applique *mutatis mutandis*. Donc, si vous avez déjà lu et compris la [section 3.17](#), vous pouvez passer à la [section 9.13](#) après la fin de cette [section](#).

Un élément plus commun est une citation ou une phrase placée juste au-dessus ou au-dessous d'un titre. Ici, la citation et la mention de la source sont généralement spécialement formatées.

<pre> \dictum[<i>auteur</i>]{<i>source</i>} \dictumwidth \dictumauthorformat{<i>auteur</i>} \dictumrule \raggeddictum raggeddictumtext \raggeddictumauthor </pre>
---

La citation peut être réglée en utilisant la commande `\dictum`. Le dicton est intégré, éventuellement conjointement avec un auteur à préciser, dans une `\parbox` (voir [\[Tea05b\]](#)) d'une largeur `\dictumwidth` mais cette longueur ne peut pas être définie par `\setlength`. Ceci est une macro qui peut être redéfinie avec `\renewcommand`.

`\textwidth` est prédéfini à 0.3333, soit un tiers de la largeur de texte. La boîte elle-même est alignée avec la commande `\raggeddictum` dont la valeur par défaut est `\raggedleft`, donc justifiée à droite. `\raggeddictum` est une macro qui peut être redéfinie en utilisant `\renewcommand`.

A l'intérieur du boîtier la citation est réglée en utilisant `\raggeddictumtext`. Le réglage par défaut est `\raggedright`, soit justifié à gauche, et il peut être redéfini avec `\renewcommand`. La sortie de la citation est fixée avec les instructions par l'élément de police qui peut être modifié par `\setkomafont` et `\addtokomafont` (voir la [section 9.6](#)). Pour le réglage par défaut, se référer à la table ?? page ??, Si un droit d'auteur est spécifié, il est séparé par une ligne sur toute la largeur `\parbox` de la maxime. Cette ligne est définie dans `\dictumrule`. Elle implique un objet vertical :

```

\cott{newcommand*\{\co{dictumrule}\}\{\co{vskip-1ex}\co{hrulefill}}
\co{par}\}}

```

qui est prédéfini.

Avec `\raggeddictumauthor` l'alignement de la ligne et de l'auteur est fait.

Par default c'est `\raggedleft`.

Cette commande peut être redéfinie avec `\renewcommand`. La sortie est sous une forme définie par `\dictumauthorformat`. La macro attend l'auteur comme argument. Par défaut c'est `\dictumauthorformat` qui définit comme :

```
\newcommand*{\dictumauthorformat}[1]{(#1)}
```

L'auteur est donc placé entre parenthèses. L'élément `dictumauthor` peut dans ce cas être défini par l'élément `dictum` dans une police différente. Pour le réglage par défaut, se référer à la table 3.16 page 209. Un changement est possible en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir section 9.6).

## 9.13 Listes

Ce qui est écrit dans l'article 2.18 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris l'article 2.18, vous pouvez continuer à la section 9.14. Toutefois, dans le paq `scrextend` seul les environnements `labeling`, `addmargin` et `addmargin*` sont pris en charge par `scrextend`. Toutes les autres listes d'environnements sont maintenues, donc laissées au contrôle de la classe utilisée.

$\LaTeX$  et les classes `standard` offrent différents environnements pour les listes. Ils sont inclus bien sûr dans `KOMA-Script`, avec parfois de légères modifications ou extensions. Fondamentalement, les listes - même de différents types - peuvent être imbriquées à une profondeur de quatre listes. Une imbrication plus profonde n'aurait, en typographie, aucun sens parce que, à proprement parler, plus de trois niveaux peut déjà être considéré comme excessif et je recommande en pareil cas de diviser une grande liste en petites listes. Puisque les listes sont des éléments standard de  $\LaTeX$ , les exemples ont été omis dans cette section. Vous pouvez en trouver section 2.18 ou dans toute présentation de  $\LaTeX$ .

```
\begin{labeling}[séparateur]{mot-clé le plus long}
\item[mot-clé] ...
:
\end{labeling}
```

Une autre forme existe dans de nombreuses classes en tant que liste de mots-clés de description d'environnement disponible uniquement dans les classes `KOMA-Script` et `scrextend` : l'environnement `labeling`.

Contrairement à la description de l’environnement précédemment présenté, un modèle peut être spécifié dans `labeling` qui détermine l’indentation de tous les éléments. Un séparateur en option peut être réglé entre l’article et le texte de description. La police pour mettre en évidence le mot-clé de l’élément `labelinglabel`, peut être modifiée en utilisant les instructions `\setkomafont` et `\addtokomafont` et rectifiée avec `labelingseparator` qui est compétent (voir la section 9.6 et la table 3.2).

L’idée de l’environnement était à l’origine pour des mises en oeuvre telles que « condition, déclaration, preuve » ou « cas, recherche, solution » qui figurent fréquemment dans les notes de cours. Depuis, il trouve des applications très différentes. Ainsi, les exemples de ce manuel sont définis avec l’environnement `labeling`.

```
\begin{addmargin}[entrée gauche]{entrée}...\end{addmargin}
\begin{addmargin*}[indentation interne]{entrée}...\end{addmargin*}
```

Comme avec l’environnement existant dans les classes `standard` et les classes `KOMA-Script`, `quote` et `quotation` utilisent l’environnement `addmargin` qui gère la marge. À la différence des deux premiers environnements, l’utilisateur peut choisir avec `addmargin` la valeur qui peut être modifiée. L’environnement ne change ni l’indentation, ni l’espace des paragraphes et aucun espace vertical supplémentaire n’est inséré après l’environnement.

Si seul l’argument obligatoire `indentation` est donné, le contenu de l’environnement à droite et à gauche est tracé avec cette valeur. Si l’argument optionnel `indentation gauche` est donné, le contenu s’écarte donc à gauche, sur la gauche, d’une valeur de retrait ajoutée à la marge.

La version étoilée `addmargin*` ne diffère que dans les présentations recto-verso, la différence se produit uniquement lorsque l’argument optionnel `indentation interne` est utilisé. Ici, la valeur de l’empreinte interne est ajoutée à la partie du bord intérieur de la page, du côté gauche de la page, sur le côté gauche, mais sur le bord droit de la page du côté à droite. L’entrée est valable pour chaque bord respectif.

Dans les deux variantes d’environnement, les valeurs négatives sont autorisées pour tous les paramètres. L’environnement se positionne ensuite en conséquence dans la marge.

Déterminer de manière fiable la position de la page, côté gauche ou droit, ne peut se faire lors de la première exécution de  $\LaTeX$ . Voir les instructions `\ifthispageodd` (section 9.8) et `\ifthispagewasodd` (article 21.1).

Dans l'interaction des environnements tels que les listes des paragraphes, le profane se pose fréquemment les mêmes questions. Par conséquent, l'explication supplémentaire de l'option 21.1 dans l'article 21.1 aborde également ce sujet. Dans la partie expert, l'article 21.1, traite aussi le problème de l'environnement `\addmargin*`.

## 9.14 Notes de marge

Ce qui est écrit à la section 3.21 s'applique *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 3.21, vous pouvez continuer au chapitre 11.

Outre l'empage, zone du texte principal qui remplit l'espace d'impression, des colonnes paramétrables dites colonnes marginales, dont il est fait ici usage, existent dans les documents.

```
\marginpar[note de marge gauche]{note de marge}  
\marginline{note de marge}
```

La commande `\marginpar`, fournie dans L<sup>A</sup>T<sub>E</sub>X, gère normalement les notes de marge. La note est placée dans le bord extérieur de la page, à droite du texte. La commande `\marginpar` permet une mise en page différente, en inversant éventuellement le côté pour utiliser la marge de gauche. Les notes de marge sortent toujours justifiées. L'expérience montre que la justification à gauche est souvent plus lisible. KOMA-Script propose la commande `\marginline`.

Un exemple détaillé peut être trouvé dans la section 3.21.

Pour les experts, les problèmes rencontrés lors de l'utilisation de `\marginpar` sont documentés et décrits dans la section 21.1. Ils s'appliquent également à `\marginline`. En outre, un paq qui permet également de gérer des notes dans des colonnes réalisées avec sa propre pagination, est présenté au chapitre 19.



# Chapitre 10

## Soutien à la pratique du droit

### 10.1 Préalable

Ce chapitre est composé de deux éléments particuliers :

- la version de 2014 due au Dr Alexander WILLAND,
- la version 2016 de Markus KOHM, intégrée au présent ouvrage.

L'approche originale du sujet par Markus KOHM m'apparaît comme un réel complément aux explications du Dr Alexander WILLAND et c'est la raison pour laquelle j'ai délibérément choisi d'intégrer les deux versions qui ne me semblent pas redondantes.<sup>1</sup>

#### 10.1.1 Généralités sur *scrjura*

Un chapitre sur le paq *scrjura* figure dans [KM12], mais à ce jour, je n'ai trouvé personne ayant une connaissance du vocabulaire juridique anglais pour traduire ce chapitre. Il serait néanmoins injuste que les utilisateurs qui ne parlent pas la langue allemande renoncent aux instructions contenues dans ce chapitre.

---

1. PAPIRAY précise : ce n'est que mon avis, je n'ai pas toujours raison et sur ce point j'ai même un témoin : demandez à mon épouse!

Il existe la présentation<sup>2</sup> libre du Dr Alexander WILLAND<sup>3</sup> (avec quelques ajouts par votre serviteur<sup>4</sup>) accessible ci-après, mais dont ni l'exhaustivité, ni la pertinence de mise à jour, ne sont garanties.

## 10.1.2 Contenu

Le paq `scrjura`, avec toutes les réserves qui s'imposent, destiné à fournir une aide dans l'évaluation et la rédaction des textes pour les avocats et les notaires, contrats flexibles, statuts et commentaires juridiques y compris la jurisprudence, est toujours en cours de développement.

## 10.1.3 Remarques juridiques

Cette partie de KOMA-Script inclut les fichiers `scrlogo.dtx` et `scrjura.dtx` selon les termes de la licence publique du projet L<sup>A</sup>T<sub>E</sub>X depuis la version 1.3c ou, à votre choix, toute version distribuée sous cette licence disponible à :

<http://www.latex-project.org/lppl.txt>

y compris dans une version de L<sup>A</sup>T<sub>E</sub>X modifiée ou ultérieure.

Le paq `scrjura` peut être diffusé indépendamment de KOMA-Script, mais la distribution du fichier `scrjura.dtx` seul sans le `scrlogo.dtx` n'est pas autorisée. Les instructions sur le paq `scrjura`, jusqu'à présent, n'existaient que sous une forme courte en langue allemande.

Le paq complet KOMA-Script se trouve sous CTAN :

`/macros/latex/contrib/KOMA-Script/`

où CTAN, synonyme abrégé pour l'annuaire `tex-archives`, regroupe tous les serveurs CTAN et les miroirs, par exemple `ftp.dante.de/tex-archive/`.

Pour consulter la liste des serveurs et miroirs CTAN :

<http://www.ctan.org>.

## 10.2 Le paq `scrjura` version 2014

---

2. Ce fichier est, au 21 juin 2017, la version `scrjura.dtx v0.9i` du 23 février 2017.

3. Auteur du manuel en langue allemande, [mailto : alexander.willand@online.de](mailto:alexander.willand@online.de)

4. Markus KOHM, auteur du paq KOMA-Script, [mailto : komascript@gmx.info](mailto:komascript@gmx.info)

### 10.2.1 Aperçu

Le paq `scrjura` est principalement destiné à l'écriture de contrats. Des options permettent d'en modifier les fonctions et le contrat lui-même est modulable, grâce à une série de commandes disponibles.

Les documents juridiques sont souvent rédigés sous une grande pression, ce qui entraîne parfois des erreurs. Ici, nous n'aurons pas besoin des messages d'erreur de  $\text{\LaTeX}2_{\epsilon}$  car `scrjura` ne nécessite, heureusement, que le paq de `scrkbase`.

Deux modes sont proposés par `scrjura` :

- le mode `contract` où le document entier est défini comme un contrat,
- le mode `environnement` : seul le texte de l'environnement `contract` est défini comme un contrat.

Le mode `contract`, activé en permanence, est déterminé par l'option `contract` choisie lors du chargement du paq.

Avec `\usepackage[contract]{scrjura}` le texte est défini comme un contrat après `\begin{document}`. Le mode environnement démarre comme n'importe quel autre mode avec `\begin{contract}` et se termine, en conséquence avec `\end{contract}`. Les deux modes ne peuvent pas être utilisés dans le même document.

Une table des matières est générée avec `\tableofcontents` et comprend, en plus des autres entêtes, tous les titres de sections, à la condition que `scrjura` charge l'option `juratotoc`. Les rubriques automatiques sont également possibles.

Chaque paragraphe a un titre, par exemple : `\Clause{title=propriété}`, dont la numérotation est automatique mais qui peut être également, effectuée manuellement.

Presque tout doit être spécifié par l'utilisateur, les données considérées comme titre de paragraphe suivent le schéma `clé=valeur`. Si un paragraphe contient plusieurs `alinéas`, tous les alinéas sont automatiquement numérotés, numérotation qui peut occasionnellement être désactivée. Il est également possible d'agir sur la composition d'un paragraphe ou d'un `alinéa` en les numérotant.

Paragraphes, `alinéas` et compositions peuvent être référencés. Pour les `alinéas` et les compositions, il y a un choix sur la manière d'exécuter la sortie ; il est possible d'obtenir, à la fois « 16 alinéa 3 composition 1 » et « 16 III 1 ».

Le paq `hyperref` est soutenu par `scrjura`, mais n'a pas été testé en détail. Comme toujours, `hyperref` est chargé en dernier. Sinon, il provoque des messages d'erreur. Si `hyperref` est retiré, le fichier `.aux*` doit être supprimé.

## 10.2.2 Options pour charger le paq

**contract** L'option `contract` assure que l'ensemble du document est interprété comme un contrat. Aucun autre environnement contractuel ne sera utilisé dans le document. Le même comportement peut être atteint en débutant l'environnement de `contract` immédiatement après `\begin{document}` que l'on ferme immédiatement avant `\end{document}`.

**juratitlepagebreak** Bien que les sauts de page ne soient pas admis dans la rubrique titre des paragraphes, certains avocats utilisent des titres si longs que même cette pagination peut y être utile. Ceci est rendu possible avec l'option `juratitlepagebreak`.

Cette option est désactivée à l'aide de `\KOMAOPTIONS{juratitlepagebreak}` à tout moment ou réactivée avec `\KOMAOPTIONS{juratitlepagebreak=false}`.

**juratotoc juratoclevel** L'option `juratotoc` contrôle et assure que les paragraphes sont entrés dans la table des matières. Dans ce cas le niveau de détail deux est utilisé par défaut pour ces entrées. Un niveau différent peut être spécifié au compteur `juratoclevel`. Le paq utilise ici les routines de `LATEX` pour créer une table des matières. Un numéro de paragraphe à trois chiffres dans la table des matières peut projeter le numéro de page dans le texte d'entête.

Le traitement le plus simple de cette erreur consiste à élargir par exemple la boîte contenant le numéro de classement en utilisant :

```
\renewcommand{\numberline}[1]{\makebox[6em][l]{#1}}
```

adapté à la largeur donnée de la boîte.

L'utilisation de `tocleft` est une autre possibilité dans une classe `KOMA-Script`, et vous pouvez bien sûr utiliser les différents paqs qui permettent d'adapter le contenu de la table des matières. Néanmoins, l'ajout d'options présente un risque accru de produire des erreurs.

Un élément clé des instructions de `scrjura`, crucial pour l'auteur, réside dans un système complet permettant de se référer à d'autres clauses du contrat.

Lors du chargement du paq, une option peut décider que les références doivent être formatées : longue (par exemple « § 314 alinéa 2 composition 2 »), courte (« § 314 al.2 p.2 ») ou numérique (« § 314 II 2 »). Les options pour cela sont :

- `ref=long`
- `ref=short`
- `ref=numeric`

Si aucune spécification n'est précisée, la valeur par défaut sera `ref=long`.

Si vous formulez des demandes spécifiques ici, la citation des paragraphes et des compositions peut être fixée différemment même dans les options. Toute combinaison de `parlong`, `parshort`, `parnumeric`, `nopar` avec `sentenceslong`, `sentencesshort`, `sentencesnumeric`, `nosentence` est possible.

Par exemple :

```
\usepackage [ref=parlong, ref=sentenceshort]{scrjura}.
```

Cependant, nous n'avons pas testé toutes ces combinaisons.

### 10.2.3 Options de titre de colonne

#### paragraphmark

Pour les contrats à long terme en cours d'exécution, les rubriques automatiques peuvent être utiles. Toutefois, la question dépend ici de la façon dont elles sont générées et *scrjura* aide à la gestion des titres courants.

Les options de `paragraphmark` peuvent être réglées sur les valeurs suivantes :

- `false`, `off`, `no`
- `right`
- `both`

Si le contrat doit être divisé en sections, donc sur une ligne, seront mentionnés `\parnumberfalse\section{paragraphe}\parnumbertrue`, sur les pages impaires des titres de `section` et les pages paires du titre du `paragraphe` courant.

En outre, les paqs `standard` peuvent être utilisés pour la conception d'entête (par exemple `scrpage2`).

## 10.2.4 Le paragraphe

`scrjura` présente deux niveaux de détail : `paragraph` et `subparagraph`.

Bien entendu, les commandes décrites dans les ouvrages suivants sont valables tant dans le mode de voisinage que dans le mode contrat. Il convient de noter que, dans les versions précédentes de commandes et de compteur `scrjura`, l'expression `paragraph` n'apparaît pas, à ce stade, dans le nom. Cependant, puisque ce terme pour les paragraphes semble inhabituel en anglais juridique, et surtout, pour éviter toute confusion avec le commandement de la division `\paragraph` normal, cela a été changé dans la version 0.7b en `Clause`. Les documents plus anciens doivent donc être ajustés en conséquence<sup>5</sup>.

`\Clause`

contient plusieurs arguments optionnels séparés par des virgules, précisés dans une liste `key=value-liste`, disponibles pour l'entête de `section`, et qui sont :

TABLE 10.1 – Liste des arguments optionnels

Entrée	Explication	Supprimer l'entrée
<code>title = &lt;Titre&gt;</code>	titre du paragraphe	<code>dummy</code>
<code>head = ...</code>	titre courant du paragraphe	<code>nohead</code>
<code>entry = ...</code>	entrée dans le contenu	<code>noentry</code>
<code>number = ...</code>	numéro du paragraphe	-

Si `\Clause{title=type d'infraction, numéro=244}` donne la sortie « § 244 type d'infraction ». `\Clause{title=type d'infraction}` donnera la sortie « § 1 type d'infraction » comme premier paragraphe, tandis que `\Clause` produit dans les mêmes conditions « § 1 », c'est-à-dire un paragraphe recensé sans titre.

Mais si vous voulez partir, par exemple, avec « § 244 » mis sur un certain nombre de paragraphes, il n'est pas conseillé d'utiliser l'option `number=` parce que cette option n'affecte pas le compteur article. Si vous ne voulez pas numéroter tous les paragraphes à la main, vous pouvez définir le point départ automa-

5. PAPIRAY précise : lors du passage en français j'ai hésité entre les termes `clause`, `paragraphe`, `sous-paragraphe` et `alinéa` qui parfois désignent ou semblent désigner le même objet. Mais si cela paraît évident pour `clause` et `paragraphe`, ainsi que pour `alinéa` et `sous-paragraphe`, comment interpréter « *Satz* » ? Cette « phrase » n'est pas toujours très claire, même si je marque ma préférence orthotypographique en l'appelant `composition`.

tique du compteur article avec `\setcounter{Clause}243`. Avec `\SubClause`, le compteur utilise naturellement un nombre en chiffres arabes, même avec une sortie ensuite en minuscules.

`\SubClause` Si un paragraphe doit être inséré, cette macro au comportement différent de `\Clause`, peut être utilisée. La numérotation démarre au numéro du paragraphe de la dernière utilisation, ajouté en lettre minuscule.

Dans `scrjura.sty`, la numérotation `\SubClause` utilise la définition :

```
\renewcommand*{\thecontractSubClause}{  
{\theClause\alph{contractSubClause}}}
```

Pour obtenir une distance entre numéro de paragraphe et lettres minuscules et une parenthèse après les minuscules, il suffit d'insérer dans le préambule :

```
\renewcommand*{\thecontractSubClause}{  
{\theClause~\alph{contractSubClause}}}
```

`\preskip \` Les distances avant et après le titre de paragraphe sont fixées avec `\setkeys{contract}{preskip=..., postskip=...,}`. Il est possible de diviser `contract` en sections en utilisant par exemple `\section`. Cependant, ce doit être complété avant la fin de l'environnement `contract` et continué par la suite (le numéro du paragraphe n'est pas réinitialisé), ou il sera précédé par `\parnumberfalse` puis par `\numbertrue`. Si vous l'oubliez, la section devient le paragraphe du premier alinéa, puis une section numérotée.

`\ellipsispar \parellipsis` Parfois il est spécialement souhaité d'omettre des alinéas dans les paragraphes, ces alinéas sont toujours comptabilisés et pris en compte dans le paragraphe avec numérotation automatique. Les alinéas oubliés sont marqués habituellement par des points de suspension, réalisés à l'aide de `\ellipsispar`. Une option peut spécifier le nombre d'alinéas qui ont été omis. Sa valeur est de un par défaut. Elle est prédéfinie avec `\textellipsis`, si la déclaration existe, sinon `\dots` est utilisé.

## 10.2.5 Références

Des liens de fonctionnement fiables sont d'une grande aide dans la rédaction de contrats. Si, dans un premier projet de contrat social ou d'accord de partenariat, les compensations de départ sont incluses dans §§ 28, elles peuvent migrer, après plusieurs conversions, dans §§ 19, puis toutes les références à l'indemnisation des actionnaires des différents régimes peuvent disparaître pour ceux qui prennent

leur retraite, y compris pour les licenciés lorsque ces changements interviennent après.

La table suivante fournit une présentation systématique de toutes les références possibles, « L » désigne la forme complète (long), « S », la forme courte (short) et « N », la représentation numérique (numeric) de la référence.

TABLE 10.2 – Les références suivantes pour § 1, alinéa 1, composition 1, sont possibles

Code	Résultat
<code>\ref{testA}</code>	§ 1, alinéa 1, composition 1
<code>\refL{testA}</code>	§ 1, alinéa 1, composition 1
<code>\refS{testA}</code>	§ 1, al. 1 C. 1
<code>\refN{testA}</code>	§ 1, I, 1
<code>\refClause{testA}</code>	§ 1
<code>\refClauseN{testA}</code>	1
<code>\refPar{testA}</code>	al. 1
<code>\refParL{testA}</code>	al. 1
<code>\refParS{testA}</code>	C. 1
<code>\refParN{testA}</code>	I
<code>\refParN[arabic]{testA}</code>	1
<code>\refSentence{testA}</code>	composition 1
<code>\refSentenceL{testA}</code>	composition 1
<code>\refSentenceS{testA}</code>	C. 1
<code>\refSentenceN{testA}</code>	1.

Par défaut, la référence `\ref{testA}` conduit à une étiquette `\label{testA}` placée dans un paragraphe, et délivre une référence complète, y compris l'indication de composition. Maintenant, vous ne référencez pas le paragraphe et l'alinéa, mais le plus souvent seulement le paragraphe et son numéro.

Tout d'abord, l'exemple qui concerne la table :

```
\begin{contract}
\Clause{title=Exemple}
l'être humain a tort.
```



```
\label{testA}
\end{contract}
```

Il en résulte un très court paragraphe :

---

§ 1 Exemple :  
l'être humain a tort.

---

Le résultat dépend des options paramétrées de `\ref`, `\refPar` et `\refSentence`.

Ici, par comparaison avec l'incorporation de `\KOMAOPTIONS{ref=nopar}`, code et résultat :

Code	Résultat
<code>\ref{testA}</code>	§ 1 composition 1
<code>\refPar{testA}</code>	
<code>\refSentence{testA}</code>	composition 1

La même chose avec `\KOMAOPTIONS{ref=nosentence}`, code et résultat :

Code	Résultat
<code>\ref{testA}</code>	§ 1 alinéa 1
<code>\refPar{testA}</code>	alinéa 1
<code>\refSentence{testA}</code>	

Et avec `\KOMAOPTION{ref}{nopar,nosentence}`, code et résultat

Code	Résultat
<code>\ref{testA}</code>	§ 1
<code>\refPar{testA}</code>	
<code>\refSentence{testA}</code>	

Pour chacun, il existe une version étoilée, par exemple `\ref*{testA}`, qui génère un lien qui n'est pas un hyperlien si le paq `hyperref` est chargé. Si le paq n'a pas été chargé, la version étoilée crée une erreur.

## 10.2.6 Alinéas et numérotation de la composition

`\pnumbertrue` `\pnumberfalse` La numérotation automatique des alinéas est désactivée, dans les paragraphes, avec `\pnumberfalse` et réactivée avec `\pnumbertrue`.

`\Sentence` Les compositions individuelles d'alinéas peuvent être numérotées avec un chiffre en exposant, mais cela ne peut être fait automatiquement. Pour régler la numérotation, utiliser la commande `\useshorthands` qui nécessite le paq `babel` :

```
\usepackage[...]{babel}
\useshorthands'{}
\defineshorthand'S}{\Sentence\ignorespaces}
\defineshorthand'{}.}{. \Sentence\ignorespaces}
```

Les ensembles peuvent être numérotés en employant le préfixe « S ». Sinon, le point final peut être fixé avec une apostrophe « ' », de sorte que le bloc suivant soit ensuite numéroté.

## 10.2.7 Personnaliser scrjura avec Kile et sous Lyx

Dans les contrats assez longs, il est très utile d'afficher la structure et le titre de paragraphe à l'écran. Qui utilise `kile` comme éditeur, que l'*open source* soit ici remercié, peut profiter de l'auto-assistance de l'outil. Dans la version 2.0.0 de `kile` il suffit d'ajouter dans le fichier `kiledocumentinfo.cpp`, à un endroit approprié, deux rangées (chacune sans saut de ligne) :

```
m_dictStructLevel[« \Clause »]=
KileStructData(5, KileStruct::Sect, « Clause »);
m_dictStructLevel[« \SubClause »]=
KileStructData(6, KileStruct::Sect, « SubClause »);
```

Les numéros au-dessus de 7 sont une mauvaise idée. Après avoir enregistré le fichier, les trois prescriptions habituelles suivent : `configure`, `make`, `make install` (ou `checkinstall...`), `kile` affiche alors « paragraphes » et « sous-paragraphes » dans la structure du document. Un utilisateur découvrira peut-être l'alternative avec `TeXnicCenter`.

Jusqu'à ce que quelqu'un écrive une classe pour utiliser `scrjura` avec `LyX`, nous voulons encourager explicitement la recherche d'une routine qui pourrait devenir une solution lors de l'utilisation de `LyX`, dans ce qui est connu des

suppléments de L<sup>A</sup>T<sub>E</sub>X comme, par exemple, l'adaptation de la définition du sous-paragraphe :

```
\renewcommand{\subsubsection}[1]{\Clause{title= #1}}
```

Ce paragraphe « rubriques » peut être défini, dans LyX, comme une sous-section de rubriques et *scrjura* numérote déjà automatiquement les alinéas de *contract*. Les inconvénients de cette approche n'apparaissent que lorsque vous avez besoin d'un sous-paragraphe.

## 10.2.8 Environnements contractuels supplémentaires

Un utilisateur ne définit, avec *scrjura*, aucun contrat ni commentaire sur les lois individuelles, mais spécifie des œuvres dans lesquelles différents types de lois sont traités. Il est donc nécessaire qu'un paragraphe ne soit pas toujours fourni avec le même préfixe « § » mais par exemple avec « art. », « loi. », ou tout autre qui peut être appelé. En outre, un comptage indépendant des différents paragraphes est nécessaire.

`\DeclareNewJuraEnvironment`

Les environnements de textes contractuels ou juridiques indépendants nécessitent la déclaration d'instructions :

```
\DeclareNewJuraEnvironment{nom}[option]
      {début d'instructions}
      {fin d'instructions}
```

L'argument *nom* est le nom du nouvel environnement. La déclaration de début (*début d'instructions*) est toujours exécutée au début de l'environnement, comme si vous l'insérez après `\begin{nom}` et la déclaration de fin (*fin d'instructions*) sera toujours exécutée à la fin de l'environnement, tout comme si vous l'insérez juste avant (`\end{nom}`).

L'argument *option* vide désigne un nouvel environnement *contract* avec ses propres compteurs. Il est possible d'utiliser une liste [Options] séparée par des virgules. Actuellement, les [Options] suivantes sont supportées :

**Clause=assignment :**

- indique avec quel environnement l'instruction `\Clause` doit être créée. La déclaration doit s'attendre exactement à un argument d'une déclaration *contract* documentée. Pour une application correcte de cette option une connaissance avancée de la fonction interne de *scrjura* est nécessaire. En outre, les exigences relatives à la déclaration d'une version peuvent encore

changer. Par conséquent, il est actuellement recommandé de ne pas utiliser cette option.

**SubClause=assignment :**

— voir l’option `Clause`.

Toutefois pour `\SubClause` au lieu de `\Clause`, `Sentence=assignment` indique avec quelle déclaration l’environnement `\Sentence` doit être créé. La déclaration ne devrait avoir aucun argument. Dans des circonstances normales, il devrait accroître le compteur `composition` avec `\refstepcounter` puis sortir sous une forme appropriée. Il est particulièrement important de faire en sorte qu’aucune espace indésirable ne soit installée.

**ClauseNumberFormat=assignment :**

— détermine la façon dont les numéros de paragraphe de cet environnement sont formatés pour l’impression. C’est une déclaration avec exactement un argument prévu, le numéro du paragraphe. Au cas où ce nombre est le dernier argument d’une chaîne d’instructions, cette chaîne peut être donnée directement comme instruction.

Par exemple, définir d’une manière satisfaisante la chaîne mentionnée ci-dessus dans l’introduction de la section d’un environnement produit :

```
\DeclareNewJuraEnvironment{article}
[ClauseNumberFormat=Art.]{}
```

Si l’élément dans une classe `KOMA-Script` est défini avec espacement au lieu d’un retrait de paragraphe, on peut utiliser :

```
\DeclareNewJuraEnvironment{article}
[ClauseNumberFormat=Art.~]
{\KOMAoptions{parskip}}{}
```

le préfixe « Art. » sera alors référencé automatiquement au lieu de « § ».

### 10.2.9 Et encore ...

`\parname`

Noms d’alinéa et de composition dépendant de la langue.

`\sentencename \thecontractClause \thecontractSubClause`

Sortie du compteur automatique de paragraphe et insertion d’un paragraphe pour `contract`.

`\AutoPar`

Numéros de `section` automatique et manuel.

`\ManualPar \Par`

Les numéros de paragraphe allumer et éteindre (`on` et `off`).

`\ifparnumber \parnumbertrue \parnumberfalse`

Compteur pour les numéros de paragraphe avec sa sortie, son formatage et le formatage de la référence. Note : si `\thepar` n'est pas seulement un chiffre arabe, les paragraphes sont numérotés avec la numérotation automatique, indépendamment de savoir s'il y a un ou plusieurs paragraphes dans la `section`.

`par  
\thepar \parformat \parciteformat`

Quelque chose comme `\ref` uniquement pour le paragraphe, pour l'`alinéa` ou pour la composition.

`\refClause`

Compteur pour le nombre composition et son impression, ainsi que le formatage de la référence.

`\refPar  
\refSentence sentence \thesentence \sentenceciteformat`

Note : `\thesentence` doit rester inchangé, Il est indispensable que `\sentenceciteformat` soit en chiffres arabes.

`\newmaxpar \getmaxpar`

Sont utilisés en interne et doivent donc être documentés pour les auteurs de paqs.

### 10.2.10 Problèmes connus

La technique de référence n'a pas été testée en détail, des erreurs sont donc possibles.

### 10.2.11 Mise en œuvre

```
1 \@ifpackageloaded{hyperref}{%  
2 \PackageError{scrjura}{Package hyperref déjà chargé}{%
```

```
3 Si vous souhaitez utiliser le package scrjura avec le package
4 hyperref, vous devez utiliser\MessageBreak
5 le package scrjura avant hyperref.\MessageBreak
6 Il vous suffit de placer le chargement du package
7 hyperref\MessageBreak
8 derrière le chargement du package scrjura.}%
9}
```

ce qui nécessite le paq `scrkbase` car tout le reste s'appuie sur lui.

```
10 \RequirePackage {scrkbase} [2013/03/26]
```

En outre est nécessaire parce que quelques commandes peuvent être utilisées.

```
11 \RequirePackage{tocbasic}
```

`contract` avec l'option `contract`, l'ensemble du document devient un contrat.

Dans ce document, on ne peut alors plus utiliser n'importe quel environnement mis à disposition par ce paq. Ceci est exécuté à partir de `\begin{document}` ou encore à partir de `\contract`.

```
12 \DeclareOption{contract}{%
13 \g@addto@macro\document\contract%
14 }
```

`juratotoc` cette option active le compteur

`juratoclevel` avec une valeur fixe souhaitée ou 2 ou 10 000.

`\if@juratotoc juratoclevel \toclevel@cpar` et le compteur s'assure à nouveau que les paragraphes sont bien ajoutés à la table des matières.

```
15 \newcounter{juratoclevel}\setcounter{juratoclevel}{\@M}
16 \KOMA@key{juratotoc}[true]{%
17 \begingroup
18 \KOMA@set@ifkey{juratoclevel}{@tempswa}{#1}%
19 \ifx\FamilyKeyState\FamilyKeyStateProcessed
20 \if@tempswa
21 \setcounter{juratoclevel}{2}%
22 \else
23 \setcounter{juratoclevel}{\@M}%
24 \fi
25 \else
26 \setcounter{juratoclevel}{#1}%
```

```

27 \FamilyKeyStateProcessed
28 \fi
29 \endgroup
30 }
31 \providecommand*\toclevel@cpa{\arabic{juratoclevel}}

```

```

juratocnumberwidth
juratocindent
\cpar@numberwidth
\cpar@indent

```

La collecte et la largeur du nombre d'entrées de répertoire peuvent être déterminées à partir de ces deux options et des macros.

```

32 \newcommand*\cpar@numberwidth{2em}
33 \newcommand*\cpar@indent{1.5em}
34 \KOMA@key{juratocnumberwidth}{%
35 \begingroup\setlength{\@tempdima}{#1}\endgroup
36 \renewcommand*\cpar@numberwidth{#1}%
37 \FamilyKeyStateProcessed
38 }
39 \KOMA@key{juratocindent}{%
40 \begingroup\setlength{\@tempdima}{#1}\endgroup
41 \renewcommand*\cpar@indent{#1}%
42 \FamilyKeyStateProcessed
43 }

```

`juratitlepagebreak` avec l'option, le commutateur `\if@juratitlepagebreak` est réglé sur `\iftrue`.

`\if@juratitle-pagebreak` ce commutateur à son tour assure qu'un saut de page est autorisé dans les rubriques des paragraphes.

```

44 \KOMA@ifkey{juratitlepagebreak}{@juratitlepagebreak}

```

`parnumber` la numérotation automatique des paragraphes peut désormais utiliser l'option `on` et `off`.

```

45 \newif\ifparnumber
46 \KOMA@key{parnumber}[true]{%
47 \ifstr{#1}{auto}{\AutoPar\FamilyKeyStateProcessed}{%
48 \ifstr{#1}{manual}{\ManualPar\FamilyKeyStateProcessed}{%

```

```

49 \KOMA@set@ifkey{parnumber}{parnumber}{#1}%
50 }%
51 }%
52 }

```

<pre> paragraphmark markright markboth \Paragraphmark </pre>
--

Avec ces options `\Paragraphmark` est redéfini afin qu'il effectue soit un `\markright` soit un `\markboth`. Contrairement à `\chaptermarkformat`, etc., `\Paragraphmark` peut être utilisé pour placer un numéro sur le côté du titre (avec modification possible si nécessaire.)

```

53 \newcommand*{\Paragraphmark}[1]{
54 \KOMA@key{paragraphmark}{%
55 \begingroup
56 \KOMA@set@ncmdkey{paragraphmark}{@tempa}{%
57 {false}{0},{off}{0},{no}{0},%
58 {forceright}{1},%
59 {forceboth}{2},%
60 {right}{3},%
61 {both}{4}%
62 }{#1}%
63 \ifx\FamilyKeyState\FamilyKeyStateProcessed
64 \ifcase\number \@tempa
65 \endgroup
66 \let\Paragraphmark@gobble
67 \or
68 \endgroup
69 \renewcommand*{\Paragraphmark}[1]{%
70 \markright{\csname MakeMarkcase\endcsname{##1}}}%
71 \or
72 \endgroup
73 \renewcommand*{\Paragraphmark}[1]{%
74 \markboth{\csname MakeMarkcase\endcsname{##1}}%
75 {\csname MakeMarkcase\endcsname{##1}}}%
76 \or
77 \endgroup

```



```
78 \renewcommand*{\Paragraphmark}[1]{%
79 \ifx
80 \@mkboth\@gobbletwo
81 \else
82 \markright{\csname MakeMarkcase\endcsname{##1}}%
83 \fi}%
84 \or
85 \endgroup
86 \renewcommand*{\Paragraphmark}[1]{%
87 \@mkboth{\csname MakeMarkcase\endcsname{##1}}%
88 {\csname MakeMarkcase\endcsname{##1}}}%
89 \else
90 \endgroup
91 \fi
92 \else
93 \endgroup
94 \fi
95 }
96 \KOMA@DeclareDeprecatedOption[scrjura]{markright}
      {paragraphmark=forceright}
97 \KOMA@DeclareDeprecatedOption[scrjura]{markboth}
      {paragraphmark=forceboth}
```

ref parcitename sentencecitename
-------------------------------------

La forme de la phrase et la forme des alinéas peuvent être référencées. Il existe trois formes : longue, courte et numérique.

<code>\parcite@format \sentencecite@format</code>
---

Le format long est utilisé par défaut. La règle suivante s'applique : 0=long, 1=court, 2=numérique, -1=rien du tout

```
98 \newcommand*{\parcite@format}{0}
99 \newcommand*{\sentencecite@-
format}{0}
```

Par défaut, l'option peut être modifiée.

```
100 \KOMA@key{ref}{%
```

```
101 \begingroup
102 \KOMA@set@ncmdkey{ref}{@tempa}{%
103 {parlong}{1},{longpar}{1},{ParL}{1},%
104 {parshort}{2},{shortpar}{2},{ParS}{2},%
105 {parnumeric}{3},{numericpar}{3},{ParN}{3},%
106 {paroff}{4},{nopar}{4},%
107 {sentencelong}{10},{longsentence}{10},{SentenceL}{10},%
108 {sentenceshort}{20},{shortsentence}{20},{SentenceS}{20},%
109 {sentencenumeric}{30},{numeralsentence}{30},{SentenceN}{30},%
110 {sentenceoff}{40},{nosentence}{40},%
111 {long}{11},%
112 {short}{22},%
113 {numeric}{33},%
114 {paragraphonly}{44},{onlyparagraph}{44},%
115 {ParagraphOnly}{44},{OnlyParagraph}{44}%
116 }{#1}%
117 \ifx\FamilyKeyState\FamilyKeyStateProcessed
118 \@tempcnta=\@tempa\relax
119 \@tempcntb=\z@
120 \@whilenum \@tempcnta>9 \do{%
121 \advance\@tempcnta -10\relax
122 \advance\@tempcntb \@ne\relax
123 }%
124 \ifcase \@tempcnta
125 \or
126 \aftergroup\def\aftergroup\parcite@format
127 \aftergroup{\aftergroup0\aftergroup}%
128 \or
129 \aftergroup\def\aftergroup\parcite@format
130 \aftergroup{\aftergroup1\aftergroup}%
131 \or
132 \aftergroup\def\aftergroup\parcite@format
133 \aftergroup{\aftergroup2\aftergroup}%
134 \or
135 \aftergroup\def\aftergroup\parcite@format
136 \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
137 \fi
138 \ifcase \@tempcntb
139 \or
```

```

140 \aftergroup\def\aftergroup\sentencecite@format
141 \aftergroup{\aftergroup0\aftergroup}%
142 \or
143 \aftergroup\def\aftergroup\sentencecite@format
144 \aftergroup{\aftergroup1\aftergroup}%
145 \or
146 \aftergroup\def\aftergroup\sentencecite@format
147 \aftergroup{\aftergroup2\aftergroup}%
148 \or
149 \aftergroup\def\aftergroup\sentencecite@format
150 \aftergroup{\aftergroup-\aftergroup1\aftergroup}%
151 \fi
152 \fi
153 \endgroup
154 }
155 \KOMA@DeclareDeprecatedOption[scrjura]{parcitename}
      {ref=parlong}
156 \KOMA@DeclareDeprecatedOption[scrjura]{sentencecitename}
      {ref=sentencelong}

```

Options d'exécution.

```
157 \KOMAProcessOptions\relax
```

`\jura@env@type` Cette macro est définie en fonction de l'environnement actif de *scrjura*.

```
158 \newcommand*{\scrjura@env@type}{}

```

`\ellipsispar \parellipsis` Mais elle omet les paragraphes avec compteur.

```

159 \newcommand*{\ellipsispar}[1][1]{%
160 \beginngroup
161 \KOMAoptions{parnumber=manual}\parellipsis\par
162 \addtocounter{par}{#1}%
163 \protected@write\@auxout{}{%
164 \string\newmaxpar{\jura@env@type}%
165 {\thecontractAbsoluteParagraph}%
166 {\thepar}%
167 }%
168 \endgroup
169 \addtocounter{par}{-1}\refstepcounter{par}%

```

```

170 \ignorespaces
171 }
172 \newcommand*\parellipsis}{%
173 \scr@ifundefinedorrelax{textellipsis}{\dots}{\textellipsis}%
174 }

```

contract
\contract
\contractParagraph
\thecontractParagraph
\contract@Paragraphformat
\Paragraphformat
contractSubParagraph
\thecontractSubParagraph
contractAbsoluteParagraph

L'environnement `contract` défini par la commande `\contract` doit seulement être utilisé comme environnement, ne peut être imbriqué, peut se terminer puis être utilisé à nouveau. Les paragraphes de `contract` sont numérotés dans le temps durablement. La fin d'un `contract` n'est donc seulement, à proprement parler, qu'une interruption.

```

175 \newcommand*\contract}{%
176 \ifx\jura@env@type\@empty
177 \let\@doendpe\contract@doendpe
178 \let\Paragraph\contract@paragraph
179 \let\c@Paragraph\c@contractParagraph
180 \edef\cl@Paragraph{\cl@Paragraph\cl@contractParagraph}%
181 \let\SubParagraph\contract@subparagraph
182 \let\c@SubParagraph\c@contractSubParagraph
183 \edef\cl@SubParagraph{\cl@SubParagraph
      \cl@contractSubParagraph}%
184 \let\Par\contract@everypar
185 \let\Sentence\contract@sentence
186 \renewcommand*\jura@env@type}{contract}%
187 \aliaskomafont{Paragraph}{contract.Paragraph}%
188 \else
189 \PackageError{contract}{nested ‘‘contract detected}{%
190 You may not use a ‘‘contract environment inside\MessageBreak
191 a ‘\jura@env@type environment or after loading\MessageBreak
192 package ‘‘scrjura with option ‘\jura@env@type!}%

```

```

193 \fi
194 }
195 \let\if@jura@skiphyperref\iftrue
196 \let\cl@Paragraph@empty
197 \let\cl@SubParagraph@empty
198 \newcounter{contractParagraph}
199 \renewcommand*{\thecontractParagraph}{%
200 {\contract@Paragraphformat{\arabic{contractParagraph}}}}
201 \DeclareRobustCommand*{\contract@Paragraphformat}[1]
      {\Paragraphformat{#1}}
202 \newcommand*{\Paragraphformat}[1]{\S~#1}
203 \newcounter{contractSubParagraph}
204 \@addtoreset{SubParagraph}{Paragraph}
205 \renewcommand*{\thecontractSubParagraph}{%
206 {\theParagraph\alph{contractSubParagraph}}}
207 \newcounter{contractAbsoluteParagraph}

```

`\contract@paragraph`

Voici la macro qui fournit à `contract` des `\paragraph` disponibles. Un contrat se compose de plusieurs paragraphes. Chaque section comporte des éléments facultatifs qui sont commandés (par l'intermédiaire de `scrkbase`) via le paq `keyval`.

```

title
head
entry tocentry nohead noentry notocentry

```

Le titre, le titre de colonne et l'entrée du répertoire du paragraphe. Le titre s'expose en premier, puis, dans l'ordre, les deux autres qui peuvent être vides. Vous pouvez aussi désactiver ce titre sans option.

```

208 \define@key{contract}{title}{%
209 \def\contract@title{#1}%
210 \ifx\contract@entry\relax\def\contract@entry{\contract@title}
      \fi
211 \ifx\contract@head\relax\def\contract@head{\contract@title}\fi
212 }
213 \define@key{contract}{entry}{%
214 \PackageWarning{scrjura}{deprecated option `entry'.
      \MessageBreak
215 You should use option `tocentry' instead of\MessageBreak

```

```

216 option `entry'%
217 }%
218 \def\contract@entry{#1}}
219 \define@key{contract}{tocentry}{\def\contract@entry{#1}}
220 \define@key{contract}{noentry}[]{%
221 \PackageWarning{scrjura}{deprecated option `noentry'.
                \MessageBreak
222 You should use option `notocentry' instead of\MessageBreak
223 option `noentry'%
224 }%
225 \let\contract@entry\relax}
226 \define@key{contract}{notocentry}[]{\let\contract@entry\relax}
227 \define@key{contract}{head}{\def\contract@head{#1}}
228 \define@key{contract}{nohead}[]{\let\contract@head\relax}

```

**number**

Le numéro peut être configuré librement avec cette option. Un paragraphe sans numérotation n'existe pas, mais si aucun chiffre n'est spécifié, le numéro sera défini automatiquement.

```

229 \define@key{contract}{number}{\def\contract@number{#1}}

```

**\contract@preskip  
\contract@postskip preskip  
postskip**

Avec ces deux options, la distance qui précède et celle qui suit un alinéa peuvent être définies. Si rien n'est spécifié, une valeur globale qui peut être réglée avec `\setkeys{contract}{.....}` est utilisée par défaut.

```

230 \newcommand*{\contract@preskip}{2\baselineskip}
231 \newcommand*{\contract@postskip}{\baselineskip}
232 \define@key{contract}{preskip}{\def\contract@preskip{#1}}
233 \define@key{contract}{postskip}{\def\contract@postskip{#1}}

```

**dummy**

Le commutateur `\ifcontract@dummy` est par défaut réglé sur `\iftrue`.

**\ifcontract@dummy**

Si le commutateur est sur `\iftrue`, le paragraphe n'est pas réglé et il est important de s'assurer que alinéas et composition du paragraphe ne peuvent

être enlevés de cette façon. Le paragraphe supprimé génère un « trou » dans la numérotation.

```
234 \newif\ifcontract@dummy
235 \define@key{contract}{dummy}[true]{\csname contract@dummy#1
        \endcsname}
```

```
\contract@paragraph@font
```

Cela doit être modifié par les éléments de l'interface de KOMA-Script.

```
236 \newkomafont{contract.Paragraph}{\sffamily\bfseries\large}
237 \newcommand*{\contract@paragraph@font}{\usekomafont{contract.
        Paragraph}%
238 \@hangfrom}
```

```
@AbsParagraph
\theH@AbsParagraph
\theHParagraph
\theHSubParagraph
```

```
239 % \textsf{hyperref} - Ce code n'a pas été testé!
240 \newcounter{@AbsParagraph}
241 \newcommand*{\theH@AbsParagraph}{P-\arabic{@AbsParagraph}}
242 \newcommand*{\theHParagraph}{\theH@AbsParagraph}
243 \newcommand*{\theHSubParagraph}{\theH@AbsParagraph}
```

Tout d'abord les numéros de section du manuel sont temporairement modifiés car aucun numéro ne doit être placé dans l'entête. Les numéros de paragraphes sont aussi modifiés puis les options sont initialisées et exécutées.

```
244 \newcommand*{\contract@paragraph}[1]{%
245 \stepcounter{contractAbsoluteParagraph}%
246 \ManualPar\parnumbertrue
247 \let\contract@title\relax
248 \let\contract@entry\relax
249 \let\contract@head\relax
250 \let\contract@number\relax
251 \contract@dummysfalse
252 \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi
```

S'il n'y a pas de paragraphe fictif, les dispositions doivent être prises pour insérer une distance verticale après le titre.

```
253 \ifcontract@dummysfalse
```

```
254 \par
255 \@afterindentfalse
256 \addvspace{\contract@preskip}%
257 \fi
```

Le nombre doit être réglé manuellement et il faut s'assurer qu'une étiquette et `hyperref` utilisent ce même numéro. Si le nombre n'est pas réglé le numéro suivant est utilisé par défaut.

```
258 \ifx\contract@number\relax
259 \let\p@Paragraph\@empty
260 \let\theParagraph\thecontractParagraph
261 \refstepcounter{Paragraph}%
262 \else
263 \begingroup
264 \let\@elt\@stpelt
265 \cl@Paragraph
266 \endgroup
267 \protected@edef\theParagraph{\contract@Paragraphformat{
                                \contract@number}}%
268 \protected@edef\@currentlabel{\theParagraph}%
269 \fi
270 \stepcounter{@AbsParagraph}%
271 \begingroup\expandafter\expandafter\expandafter\endgroup
272 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
273 \else
274 \expandafter\let\csname if@jura@skiphyperref\expandafter
                                \endcsname
275 \csname if@skiphyperref\endcsname
276 \fi
277 \if@jura@skiphyperref\else
278 \hyper@refstepcounter{@AbsParagraph}%
279 \typeout{absolute Nummer: \the@AbsParagraph^^JLabel:
                                '\@currentHref '}%
280 \fi
```

Dans le but de simplifier, le développement se fait à partir d'ici comme s'il s'agissait d'une sous-section.

```
281 \let\theSubParagraph\theParagraph
```



Le titre est ensuite fixé, sauf pour un paragraphe fictif, en fabriquant les entrées de répertoire et en adaptant le titre des colonnes.

```

282 \ifcontract@dummy\else
283 \begingroup
284 \if@juratitlepagebreak\else\interlinepenalty\@M\fi
285 \contract@paragraph@font{\theParagraph
286 \ifx\contract@title\relax\else\enskip\fi}%
287 \contract@title
288 \ifx\contract@entry\relax\else
289 \addxcontentsline{toc}{cpar}[\theParagraph]\contract@entry
290 \addxcontentsline{cpa}{cpar}[\theParagraph]\contract@entry
291 \fi
292 \ifx\contract@head\relax\else
293 \expandafter\Paragraphmark\expandafter{%
294 \expandafter\theSubParagraph\expandafter\enskip\contract@head}%
295 \fi
296 \par
297 \endgroup\nobreak\vskip\contract@postskip

```

Enfin, on contrôle que les alinéas sont numérotés correctement, etc.

```

298 \jura@afterheading{contract}%
299 \fi
300 }

```

`\contract@subparagraph`

Cela correspond à quelques petites choses se rapportant à des sous-paragraphe de `\contract@paragraph`.

```

301 \newcommand*{\contract@subparagraph}[1]{%
302 \stepcounter{contractAbsoluteParagraph}%
303 \ManualPar\parnumbertrue
304 \let\contract@title\relax
305 \let\contract@entry\relax
306 \let\contract@head\relax
307 \let\contract@number\relax
308 \contract@dummysfalse
309 \ifx\relax#1\relax\else\setkeys{contract}{#1}\fi
310 \ifcontract@dummy\else
311 \par

```

```
312 \@afterindentfalse
313 \vskip\contract@preskip
314 \fi
315 \ifx\contract@number\relax
316 \let\p@SubParagraph\@empty
317 \let\theSubParagraph\thecontractSubParagraph
318 \refstepcounter{SubParagraph}%
319 \else
320 \begingroup
321 \let\@elt\@stpelt
322 \cl@SubParagraph
323 \endgroup
324 \protected@edef\theSubParagraph{\theParagraph\contract@number}%
325 \protected@edef\@currentlabel{\theSubParagraph}%
326 \fi
327 \stepcounter{@AbsParagraph}%
328 \begingroup\expandafter\expandafter\expandafter\endgroup
329 \expandafter\ifx\csname if@skiphyperref\endcsname\relax
330 \else
331 \expandafter\let\csname if@jura@skiphyperref\expandafter
    \endcsname
332 \csname if@skiphyperref\endcsname
333 \fi
334 \if@jura@skiphyperref\else
335 \hyper@refstepcounter{@AbsParagraph}%
336 \typeout{absolute Nummer: \the@AbsParagraph^^JLabel:
    '\@currentHref}%
337 \fi
338 \ifcontract@dumy\else
339 \begingroup
340 \if@juratitlepagebreak\else\interlinepenalty\@M\fi
341 \contract@paragraph@font{\theSubParagraph
342 \ifx\contract@title\relax\else\enskip\fi}%
343 \contract@title
344 \ifx\contract@entry\relax\else
345 \addxcontentsline{toc}{cpar}[\theSubParagraph]\contract@entry
346 \addxcontentsline{cpar}{cpar}[\theSubParagraph]\contract@entry
347 \fi
348 \ifx\contract@head\relax\else
```

```
349 \expandafter\Paragraphmark\expandafter{%
350 \expandafter\theSubParagraph\expandafter\enskip\contract@head}%
351 \fi
352 \par
353 \endgroup
354 \nobreak\vskip\contract@postskip
355 \jura@afterheading{contract}%
356 \fi
357 }
```

<code>\AutoPar</code>
<code>\ManualPar</code>

Activation automatique ou manuelle des numéros de section pour tous les environnements.

```
358 \newcommand*\AutoPar{%
359 \expandafter\let\expandafter\jura@everypar
360 \csname \jura@type @everypar\endcsname
361 }
362 \newcommand*\ManualPar{%
363 \let\jura@everypar\relax
364 }
```

<code>\jura@afterheading</code>
---------------------------------

Correspond à `\afterheading` avec des extensions pour les numéros automatiques de paragraphes.

```
365 \newcommand*\jura@afterheading[1]{%
366 \@nobreaktrue
367 \everypar{%
368 \if@nobreak
369 \@nobreakfalse
370 \clubpenalty \@M
371 \if@afterindent \else
372 {\setbox\z@\lastbox}%
373 \fi
374 \else
375 \clubpenalty \@clubpenalty
376 \everypar{%
377 \jura@everypar
```

```
378 }%
379 \fi
380 \jura@everypar
381 }%
382 \def\jura@type{#1}%
383 \AutoPar
384 }
```

`\jura@everypar`

La requête qui sera exécutée au début de chaque alinéa pour en définir le numéro d'ordre. N'est pas utile en dehors des environnements.

```
385 \newcommand*\jura@everypar{}
```

`\@doendpe`

Cette instruction est utilisée par L<sup>A</sup>T<sub>E</sub>X pour réinitialiser les actions à court, moyen et long terme en agissant à la fin des environnements d'alinéas. Elle est réinsérée pour éviter que la numérotation automatique des paragraphes ne soit désactivée.

```
386 \CheckCommand*\@doendpe{\@endpetrue
387 \def\par{\@restorepar\everypar}\par\@endpefalse}\everypar
388 {\setbox\z@\lastbox}\everypar}\@endpefalse}}
389 \newcommand*\contract@doendpe{%
390 \@endpetrue
391 \def\par{%
392 \@restorepar\everypar{%
393 \csname jura@everypar\endcsname
394 }%
395 \par\@endpefalse
396 }%
397 \everypar{%
398 {\setbox\z@\lastbox}\everypar{%
399 \csname jura@everypar\endcsname
400 }%
401 \@endpefalse
402 }%
403 }
```

`\l@cpar`

Entrée de répertoire pour un paragraphe de contract.

```

404 \newcommand*\l@cpar}[2]{%
405 \ifnum\value{juratoclevel}>\value{tocdepth}\else
406 \scr@ifundefinedorrelax{bprot@dottedtocline}{%
407 \@dottedtocline
408 }{%
409 \bprot@dottedtocline
410 }{\value{juratoclevel}}{\cpar@indent}{\cpar@numberwidth}
      {#1}{#2}%
411 \fi
412 }
```

`\jura@separator`

L'espace peut être oté au début ou à la fin.

```
413 \DeclareRobustCommand*\jura@separator}[1]{#1}
```

`\jura@usetype`

Ne fait rien par défaut, il est robuste et appelle `\jura@@usetype`.

`\jura@@usetype`

En revanche, non robuste mais plus facilement redéfinissable. Par défaut, ne sert réellement pas à grand-chose sauf anéantir l'argument.

```
414 \DeclareRobustCommand*\jura@usetype}[1]{\jura@@usetype{#1}}
415 \newcommand*\jura@@usetype}[1]{}
```

`\contract@everypar`

`\jura@everpar` pour les contrats.

```

\ifparnumber
par
\thepar
\theHpar
\parformat
\p@par
```

Le commutateur spécifie si les numéros d'alinéas doivent être utilisés ou non. Les numéros sont désactivés et aucun numéro manuel n'est défini. Les paragraphes ne sont alors pas comptés. Sinon, les paragraphes sont numérotés

avec `\thepar`. Il est important que le compteur des paragraphes et alinéas soit remis à zéro. En outre, l'article doit être inclus dans la sortie et cité comme un objet parent.

```
416 \newcounter{par}
417 \renewcommand*{\thepar}{\arabic{par}}
418 \newcommand*{\theHpar}{\theH@AbsParagraph-\Roman{par}}
419 \newcommand*{\parformat}{(\thepar)}
420 \newkomafont{parnumber}{}
421 \renewcommand*{\p@par}{\jura@usetype{\jura@env@type}
      \theSubParagraph\jura@separator{\nobreakspace}}
422 \@addtoreset{par}{Paragraph}
423 \@addtoreset{par}{SubParagraph}
424 \newcommand*{\contract@everypar}{%
425 \ifparnumber
426 \ifx\scrjura@special@par\relax
427 \ifx\scrjura@special@reset@par\relax\else
428 \global\let\thepar\scrjura@special@reset@par
429 \global\let\scrjura@special@reset@par\relax
430 \fi
431 \refstepcounter{par}%
432 \refstepcounter{sentence}%
433 \else
434 \ifx\scrjura@special@reset@par\relax
435 \global\let\scrjura@special@reset@par\thepar
436 \fi
437 \global\let\thepar\scrjura@special@par
438 \global\let\scrjura@special@par\relax
439 \setcounter{sentence}{0}\refstepcounter{sentence}%
440 \fi
441 \begingroup
442 \protected@write\@auxout{\let\contract@Paragraphformat
      \@firstofone}{%
443 \string\newmaxpar{contract}%
444 {\thecontractAbsoluteParagraph}%
445 {\thepar}}%
446 \getmaxpar\@tempa{contract}{\thecontractAbsoluteParagraph}%
447 \typeout{Stored max is \@tempa}%
448 \def\reserved@a##1\@nnil{\def\@tempa{##1}}%
```

```

449 \afterassignment\reserved@a\@tempcnta=0\@tempa\relax\@nnil
450 \ifnum \@tempcnta>\@ne
451 {\usekomafont{parnumber}{\parformat\nobreakspace}}%
452 \else
453 \def\reserved@a{\relax}%
454 \ifx\@tempa\reserved@a
455 \else
456 {\usekomafont{parnumber}{\parformat\nobreakspace}}%
457 \fi
458 \fi
459 \endgroup
460 \else
461 \setcounter{sentence}{-1}\refstepcounter{sentence}%
462 \fi
463 }

```

`\thisparnumber \scrjura@special@par \scrjura@special@reset@par`

Cette commande permet l'attribution manuelle d'un numéro de paragraphe dont le chiffre doit être entièrement extensible.

```

464 \newcommand*\thisparnumber}[1]{%
465 \def\scrjura@special@par{#1}%
466 }
467 \newcommand*\scrjura@special@par{}
468 \let\scrjura@special@par\relax
469 \newcommand*\scrjura@special@reset@par{}
470 \let\scrjura@special@reset@par\relax

```

`\refL \ref@L`

La même chose que `\ref` mais dans la forme longue.

```

471 \newcommand*\refL{\@ifstar {\ref@L*}{\ref@L{}}}
472 \newcommand*\ref@L[2]{%
473 \begingroup
474 \def\parcite@format{0}%
475 \let\sentencecite@format\parcite@format
476 \ref#1{#2}%
477 \endgroup
478 }

```

`\refS \ref@S`

La même chose que `\ref` mais dans la forme courte.

```
479 \newcommand*{\refS}{\@ifstar {\ref@S*}{\ref@S{}}}  
480 \newcommand*{\ref@S}[2]{%  
481 \begingroup  
482 \def\parcite@format{1}%  
483 \let\sentencecite@format\parcite@format  
484 \ref#1{#2}%  
485 \endgroup  
486 }
```

`\refN?? \ref@N`

La même chose que `\ref??\ref` mais dans la forme numérique.

```
487 \newcommand*{\refN}{\@ifstar {\ref@N*}{\ref@N{}}}  
488 \newcommand*{\ref@N}[2]{%  
489 \begingroup  
490 \def\parcite@format{2}%  
491 \let\sentencecite@format\parcite@format  
492 \ref#1{#2}%  
493 \endgroup  
494 }
```

`\refParagraph \ref@Paragraph`

Cite seulement un paragraphe, les sous-paragraphes ou une phrase. Pour plus de compatibilité avec `hyperref` il existe une forme `hyperref` étoilée. Sans `hyperref` cela ne produit qu'un non-sens.

```
495 \newcommand*{\refParagraph}{%  
496 \@ifstar {\ref@Paragraph*}{\ref@Paragraph{}}  
497 }  
498 \newcommand*{\ref@Paragraph}[2]{%  
499 \expandafter\ifx\csname r@#2\endcsname\relax  
500 \ref#1{#2}%  
501 \else  
502 \begingroup
```

Classe toutes les parties de la référence jusqu'à la première dans `\@tempb`.

```
510 \def\@tempc##1##2\@nil{##1}%  
511 \let\jura@separator\@gobble
```



```
512 \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
513 \csname r@#2\endcsname\noexpand\@nil}%
```

Prend la première partie de @tempa dans @tempa.

```
514 \protected@edef\@tempa{\expandafter\expandafter\expandafter\@tempc
515 \@tempa\@nil}%
516 \let\@@protect\protect
517 \let\protect\noexpand
518 \expandafter\edef\csname r@#2\endcsname{\@tempa\@tempb}%
519 \let\protect\@@protect
520 \ref#1{#2}%
521 \endgroup
522 \fi
523 }
```

```
\refParagraphN \ref@ParagraphN
```

Cite seulement un paragraphe, les sous-paragraphe ou une phrase. Pour plus de compatibilité avec `hyperref` il existe une forme `hyperref` étoilée. Sans `hyperref` cela ne produit qu'un non-sens.

```
524 \newcommand*\refParagraphN{%
525 \@ifstar {\ref@ParagraphN*}{\ref@ParagraphN{}}
526 }
527 \newcommand*\ref@ParagraphN[2]{%
528 \begingroup
529 \let\Paragraphformat\relax
530 \ref@Paragraph{#1}{#2}%
531 \endgroup
532 }
```

```
\refPar \ref@Par
```

Cite seulement un alinéa d'un paragraphe ou d'une composition. La forme étoilée présente plus de compatibilité avec le paq `hyperref` sans lequel cela ne produit qu'un non-sens.

```
533 \newcommand*\refPar{%
534 \@ifstar {\ref@Par*}{\ref@Par{}}
535 }
536 \newcommand*\ref@Par[2]{%
537 \expandafter\ifx\csname r@#2\endcsname\relax
```

538 \ref#1{#2}%

539 \else

540 \begingroup

Classe dans \@tempb toutes les parties de la référence sauf la première.

541 \expandafter\expandafter\expandafter\expandafter

542 \expandafter\expandafter\expandafter\def

543 \expandafter\expandafter\expandafter\expandafter

544 \expandafter\expandafter\expandafter\@tempb

545 \expandafter\expandafter\expandafter\expandafter

546 \expandafter\expandafter\expandafter{%

547 \expandafter\expandafter\expandafter\@gobble\csname r@#2\end  
csname}%

Classe la première partie de la référence dans \@tempa.

548 \def\@tempc##1##2\@nil{##1}%

549 \let\jura@separator\@gobble

550 \protected@edef\@tempa{\expandafter\expandafter\expandafter  
\@tempc

551 \csname r@#2\endcsname\noexpand\@nil}%

Classe la deuxième partie de \@tempa dans \@tempa.

552 \def\@tempc##1##2##3\@nil{##2}%

553 \protected@edef\@tempa{\expandafter\expandafter\expandafter  
\@tempc

554 \@tempa{%

555 \protect\G@refundefinedtrue

556 \nfss@text{\reset@font\bfseries ??}%

557 \@latex@warning{Reference ‘ ’#2 on page \thepage \space

558 with undefined par number}%

559 }\noexpand\@nil}%

560 \let\@@protect\protect

561 \let\protect\noexpand

562 \expandafter\edef\csname r@#2\endcsname{{\@tempa}\@tempb}%

563 \let\protect\@@protect

564 \ref#1{#2}%

565 \endgroup

566 \fi

567 }

```
\refParL \ref@ParX
```

```
568 % L'équivalent obligatoirement long.
569 \newcommand*\refParL}{%
570 \@ifstar {\ref@ParX0*}{\ref@ParX0{}}
571 }
572 \newcommand*\ref@ParX}[3]{%
573 \begingroup
574 \def\parcite@format{#1}%
575 \let\sentencecite@format\parcite@format
576 \ref@Par{#2}{#3}%
577 \endgroup
578 }
```

```
\refParS
```

L'équivalent obligatoirement court.

```
579 \newcommand*\refParS}{%
580 \@ifstar {\ref@ParX1*}{\ref@ParX1{}}
581 }
```

```
\refParN
```

L'équivalent obligatoirement numérique.

```
582 \newcommand*\refParN}{%
583 \@ifstar {\ref@ParN2*}{\ref@ParN2{}}
584 }
585 \newcommand*\ref@ParN}[2]{%
586 \@ifnextchar [%]
587 {\ref@@ParN{#1}{#2}}%
588 {\ref@ParX{#1}{#2}}%
589 }
590 \newcommand*\ref@@ParN}{%
591 \def\ref@@ParN#1#2[#3]#4{%
592 \begingroup
593 \renewcommand*\parnumericformat}[1]{%
594 \csname @#3\endcsname{\number ##1\relax}%
595 }%
596 \ref@ParX{#1}{#2}{#4}%
597 \endgroup
598 }
```

```
\refSentence \ref@Sentence
```

Pour citer seulement l'ensemble d'un ensemble. La forme étoilée présente plus de compatibilité avec le paq hyperref sans lequel cela ne produit qu'un non-sens.

```
599 \newcommand*{\refSentence}{%
600 \@ifstar {\ref@Sentence*}{\ref@Sentence{}}
601 }
602 \newcommand*{\ref@Sentence}[2]{%
603 \expandafter\ifx\csname r@#2\endcsname\relax
604 \ref#1{#2}%
605 \else
606 \begingroup
```

Classer toutes les parties de la référence jusqu'à la première dans \@tempb.

```
607 \expandafter\expandafter\expandafter\expandafter
608 \expandafter\expandafter\expandafter\def
609 \expandafter\expandafter\expandafter\expandafter
610 \expandafter\expandafter\expandafter\@tempb
611 \expandafter\expandafter\expandafter\expandafter
612 \expandafter\expandafter\expandafter{%
613 \expandafter\expandafter\expandafter\@gobble\csname r@#2
\endcsname}%
```

Classer la première partie de la référence dans \@tempa.

```
614 \def\@tempc##1##2\@nil{##1}%
615 \let\jura@separator\@gobble
616 \protected@edef\@tempa{\expandafter\expandafter\expandafter
\@tempc
617 \csname r@#2\endcsname\noexpand\@nil}%
```

Classer la troisième partie de \@tempa dans \@tempa.

```
618 \def\@tempc##1##2##3##4\@nil{##3}%
619 \protected@edef\@tempa{\expandafter\expandafter\expandafter
\@tempc
620 \@tempa}{%
621 \protect\G@refundefinedtrue
622 \nfss@text{\reset@font\bfseries ??}%
623 \@latex@warning{Reference ‘#2 on page \thepage \space
624 with undefined sentence number}%
```

```

625 }\noexpand\@nil}%
626 \let\@@protect\protect
627 \let\protect\noexpand
628 \expandafter\edef\csname r@#2\endcsname{\@tempa}\@tempb}%
629 \let\protect\@@protect
630 \ref#1{#2}%
631 \endgroup
632 \fi
633 }

```

```
\refSentenceL \ref@SentenceX
```

```

634 % L'équivalent obligatoirement long.
635 \newcommand*\refSentenceL}{%
636 \@ifstar {\ref@SentenceX0*}{\ref@SentenceX0{}}
637 }
638 \newcommand*\ref@SentenceX}[3]{%
639 \begingroup
640 \def\parcite@format{#1}%
641 \let\sentencecite@format\parcite@format
642 \ref@Sentence{#2}{#3}%
643 \endgroup
644 }

```

```
\refSentenceS
```

L'équivalent nécessairement court.

```

645 \newcommand*\refSentenceS}{%
646 \@ifstar {\ref@SentenceX1*}{\ref@SentenceX1{}}
647 }

```

```
\refSentenceN
```

L'équivalent nécessairement numérique.

```

648 \newcommand*\refSentenceN}{%
649 \@ifstar {\ref@SentenceX2*}{\ref@SentenceX2{}}
650 }

```

```
\contract@sentence
```

Ensembles numérotés.

```
sentence
\thesentence
\theHsentence
\p@sentence
```

Le compteur est utilisé pour la numérotation des compositions. Il est important de les citer comme objets parents du paragraphe.

```
651 \newcounter{sentence}[par]
652 \renewcommand*{\thesentence}{\arabic{sentence}}
653 \newcommand*{\theHsentence}{\theHpar-\arabic{sentence}}
654 \renewcommand*{\p@sentence}{\expandafter\p@@sentence}
655 \newcommand*{\p@@sentence}[1]{\p@par{\par@cite{\theHpar}}}%
656 \jura@separator{\nobreakspace}{\sentence@cite{#1}}
657 \newcommand*{\contract@sentence}{%
658 \refstepcounter{sentence}%
659 \ensuremath{\mbox{\fontsize\sf@size\z@\selectfont
\thesentence}}}%
660 }
```

```
\parciteformat
```

Citation pour les paragraphes.

```
661 \DeclareRobustCommand*{\par@cite}[1]{\parciteformat{#1}}
662 \newcommand*{\parciteformat}[1]{%
663 \ifcase \parcite@format
664 \expandafter\parlongformat
665 \or
666 \expandafter\parshortformat
667 \or
668 \expandafter\parnumericformat
669 \else
670 \unskip\expandafter\@gobble
671 \fi
672 {#1}%
673 }
```

```
\sentenceciteformat
```

Style de citation mise au point par défaut avec \@arabic.

```
674 \DeclareRobustCommand*{\sentence@cite}[1]
{\sentenceciteformat{#1}}
```

```

675 \newcommand*{\sentenceciteformat}[1]{%
676 \ifcase \sentencecite@format
677 \expandafter\sentencelongformat
678 \or
679 \expandafter\sentenceshortformat
680 \or
681 \expandafter\sentencenumericformat
682 \else
683 \unskip\expandafter\@gobble
684 \fi
685 {#1}%
686 }

```

```

\parlongformat
\parshortformat
\parnumericformat
\sentencelongformat
\sentenceshortformat
\sentencenumericformat

```

Le formatage dans les six formes.

```

687 \newcommand*{\parlongformat}[1]{\parname~#1}
688 \newcommand*{\parshortformat}[1]{\parshortname~#1}
689 \newcommand*{\parnumericformat}[1]{\@Roman{\number #1
\relax}}
690 \newcommand*{\sentencelongformat}[1]{\sentencename~#1}
691 \newcommand*{\sentenceshortformat}[1]{\sentenceshortname~#1}
692 \newcommand*{\sentencenumericformat}[1]{\@arabic{\number #1
\relax}.}

```

```

\parname
\parshortname
\sentencename
\sentenceshortname
\scrjura@lang@error

```

Le nom d'un alinéa et d'une composition. Les noms anglais sont manipulés avec la contribution de « m.eik ».

```

693 \newcommand*{\parname}{Paragraph}
694 \AtBeginDocument{%

```

```
695 \providecaptionname{german,ngerman,austrian,naustrian}
      \parname{Absatz}%
696 \providecaptionname{german,ngerman,austrian,naustrian}
      \parshortname{Abs.}%
697 \providecaptionname{german,ngerman,austrian,naustrian}
      \sentencename{Satz}%
698 \providecaptionname{german,ngerman,austrian,naustrian}
      \sentenceshortname{S.}%
699 \providecaptionname{english,american,british,canadian,
      USenglish,UKenglish}\parname{paragraph}%
700 \providecaptionname{english,american,british,canadian,
      USenglish,UKenglish}\parshortname{par.}%
701 \providecaptionname{english,american,british,canadian,
      USenglish,UKenglish}\sentencename{sentence}%
702 \providecaptionname{english,american,british,canadian,
      USenglish,UKenglish}\sentenceshortname{sent.}%
703 }
704 \providecommand*{\parname}{\scrjura@lang@error{\parname}}
705 \providecommand*{\parshortname}{\scrjura@lang@error
      {\parshortname}}
706 \providecommand*{\sentencename}{\scrjura@lang@error
      {\sentencename}}
707 \providecommand*{\sentenceshortname}{\scrjura@lang@error
      {\sentenceshortname}}
708 \newcommand*{\scrjura@lang@error}[1]{%
709 \PackageError{scrjura}{%
710 current language not supported%
711 }{%
712 Currently scrjura only supports languages `german', `ngerman',
713 `austrian',\MessageBreak
714 `naustrian', `english', `american', `british', `canadian',
      `USenglish',
715 and\MessageBreak
716 `UKenglish'.\MessageBreak
717 It seems, that you are using another language (maybe
      `\'languagename) or
718 that\MessageBreak
719 your language selection 'isnt compatible to package `babel'.
      \MessageBreak
```



```
720 Because of this you have to define `string#1' by yourself!
      \MessageBreak
721 It would be nice if 'youll send your definitions to the author.%
722 }%
723 \textbf{??}%
724 }
```

```
\newmaxpar
\getmaxpar
```

Deux macros auxiliaires pour verrouiller le compteur dans le fichier, puis pour choisir les valeurs si elles ne sont pas définies dans le fichier .aux.

```
725 \newcommand*{\newmaxpar}[3]{%
726 \begingroup
727 \expandafter\let\csname #1@Paragraphformat\endcsname
      \@firstofone
728 \protected@edef\@tempa{#2}\@onelevel@sanitize\@tempa
729 \expandafter\xdef\csname max@#1@\@tempa\endcsname{#3}%
730 \endgroup
731 }
732 \newcommand*{\getmaxpar}[3]{%
733 \begingroup
734 \expandafter\let\csname #2@Paragraphformat\endcsname
      \@firstofone
735 \protected@edef\@tempa{#3}%
736 \@onelevel@sanitize\@tempa
737 \expandafter\ifx \csname max@#2@\@tempa\endcsname\relax
738 \edef\@tempa{\endgroup\edef\noexpand#1{\expandafter\the\value
      {par}}}%
739 \else
740 \edef\@tempa{\endgroup
741 \edef\noexpand#1{\csname max@#2@\@tempa\endcsname}}%
742 \fi
743 \@tempa
744 }
```

## 10.3 Généralités sur scrjura version 2016

Voulez-vous signer un contrat, établir les statuts d'une association ou d'une société, préparer un projet de loi ou écrire un commentaire juridique ? alors le paq `scrjura` vous est nécessaire pour cette activité. Bien que conçu comme une aide générale pour les documents d'avocats, il a été démontré que le contrat en est un élément central. Une attention particulière est portée ici au paragraphe avec numéro, titre, sous-paragraphe numérotés - s'il y a plusieurs d'entre eux dans un seul paragraphe - et si nécessaire les ensembles, entrées dans la table des matières et références croisées dans les formes appropriées de la pratique allemande. Ce paq est réalisé en coopération avec le Dr. Alexander WILLAND<sup>6</sup>, avocat établi à KARLSRUHE. Les énoncés suivants expliquent les fonctions sous un angle légèrement différent de celui des instructions du paq inclus. Il convient de noter que ce paq fonctionne avec `hyperref` qui doit, comme d'habitude, être chargé pour `scrjura`.

### 10.3.1 Choix d'option précoce ou tardive

Ce qui est écrit dans la `section 2.4` est applicable *mutatis mutandis*.

### 10.3.2 Répertoires

Le paq `scrjura` fournit également les entrées dans la table des matières.

<code>juratotoc=off-on-sélection</code>
<code>juratotoc=numéro de niveau</code>

Elles apparaissent dans la `section` contenu si leur nombre est inférieur ou égal au niveau du compteur `tocdepth` (voir `section 3.9`). Le réglage par défaut est le numéro de niveau, la valeur 10.000 est également utilisée lorsque l'option via une valeur de marche-arrêt (dans la table 1.6 page 20) est désactivée. Le compteur `tocdepth` a généralement une faible valeur à un seul chiffre, les paragraphes entrées n'apparaissent donc pas, normalement, dans la table des matières. Si d'autre part, l'option est activée avec une valeur unique (`off`, `on`, `valeur`), elle est définie comme nombre de niveau 2 dans toutes les classes KOMA-Script au même niveau que `\subsection`. Par défaut, `tocdepth` est intégré dans la table des matières de ces classes.

---

6. Voir la `section` « Généralités sur `scrjura` » version 2014.

```
juratocindent=entrée
juratocnumberwidth=largeur du numéro
```

Ces deux options peuvent être utilisées pour la collecte des entrées déterminées des paragraphes dans la table des matières et dont la largeur est réservée aux numéros. La valeur par défaut est la même que pour l'entrée de `\subsection` dans `scrartcl`.

### 10.3.3 Environnement pour les contrats

Les principaux mécanismes de *scrjura* sont destinés exclusivement aux contrats et environnements associés.

```
\begin{contract}
...
\end{contract}
```

Ceci est, à ce jour, le premier et le seul environnement pour avocats et qui est fourni par *scrjura*. L'utilisation de la numérotation automatique du paragraphe est activée et les instructions `\Clause` et `\SubClause` pourvues d'une forme spécifique sont ensuite documentées en détail. Les environnements `contract` ne doivent pas être imbriqués l'un dans l'autre, ce qui n'empêche pas d'en utiliser plusieurs dans un document. Les paragraphes sont traités dans ce cas comme s'ils étaient dans un seul environnement, y mettre fin (ou seulement l'interrompre) déclenche l'ancien qui continue comme s'il débutait un nouvel environnement, sans interruption possible à l'intérieur d'un paragraphe.

```
\contract
```

Lorsque cette option est spécifiée pour charger le paq, par exemple avec `\usepackage` ou comme une option globale avec `\documentclass`, l'ensemble du document est consacré à un contrat et devra se comporter également de telle manière avec cet environnement de contrat. Noter que les options de `contract` ne peuvent être réglées avec `\KOMAOption` ou `\KOMAOptions`. Ainsi, il est impossible de sélectionner ou de désactiver l'option à nouveau. Dans ce cas, utiliser directement un environnement de `contract`.

### 10.3.4 Section juridique

Les paragraphes<sup>7</sup> dans le sens juridique ne sont définis avec `scrjura` que dans les contrats ou dans un environnement `contract`.

<code>\Clause[réglages]</code>
<code>\SubClause[réglages]</code>

Telles sont les principales actions dans un contrat. Sans paramètres supplémentaires `\clause` crée un titre de paragraphe qui se compose uniquement du paragraphe suivi d'un numéro.

En revanche, `\subclause` crée pour `\clause` un intitulé avec numéro de l'article pour la dernière série de la lettre d'affaire en cours qui sera ajouté.

L'idée est que `\SubClause` est principalement utilisé non seulement pour le cas où, dans l'amendement des lois ou des contrats, le paragraphe doit être reformulé ou supprimé, mais aussi dans le cas où de nouveaux paragraphes sont insérés entre les paragraphes existants sans une nouvelle numérotation. Les réglages peuvent spécifier les deux états dans une liste des propriétés séparées par une virgule. Un aperçu des propriétés possibles est offert dans la table 10.2 page 322 dont les plus importantes peuvent être examinées plus en détail. Par défaut, une distance de deux lignes et un espacement d'une ligne sont insérés avant le titre. Les propriétés `preskip` et `postskip` permettent de modifier ces distances. Le nouvel ajustement vaut non seulement pour le paragraphe en cours, mais à partir du paragraphe jusqu'à la fin de l'environnement actuel. Il est également possible de modifier le réglage approprié avant l'utilisation :

```
\setkeys{contract}{preskip=distance, postskip=distance}
```

rend possible un paramétrage indépendant des paragraphes individuels, même en dehors d'un environnement `contract` ajouté dans le préambule après le chargement de `scrjura`. Cependant, il est impossible d'indiquer à ces deux options de charger le paq pour le mettre dans `\KOMAOPTIONS` ou `\KOMAOPTION`.

La configuration par défaut utilise la police `\sffamily\bfseries \large` pour le titre des paragraphes.

À propos de l'article, `contract.Clause` peut en modifier l'écriture avec les commandes `\setkomafont` et `\addtokomafont` (voir section 3.6).

---

7. Bien que le terme juridique exact soit « `section` » pour les paragraphes en anglais, la désignation « `clause` » dans `scrjura` en est le dérivé, ceci pour éviter la confusion à la fois avec `\section` et `\paragraph`.

Dans l'environnement `contract`, `clause` peut être également utilisé à la place de `contract.Clause`.

En utilisant les paramètres `title`, `head` et `tocentry`, les paragraphes peuvent être fournis avec un titre en plus du numéro.

Il est recommandé de définir la valeur de la propriété respective de chaque accolade. Sinon, des virgules, par exemple, peuvent conduire à une confusion avec le séparateur entre des propriétés différentes. Les valeurs vides pour `head` et `tocentry` provoquent des entrées vides. Cependant, pour faire des entrées, utiliser `nohead` et `notocentry`.

Le numéro de série est incrémenté automatiquement ; il peut être attribué manuellement, ce qui ne modifie pas les numéros des paragraphes suivants, et la spécification d'un numéro vide n'est pas permise. Les instructions fragiles concernant les nombres doivent être protégées par `\protect`. Pour numéroter, il est recommandé de n'utiliser que des chiffres et des lettres.

### 10.3.5 Propriétés de `dummy`

On peut utiliser la propriété `dummy` pour supprimer l'édition du titre complet du paragraphe. Dans le comptage automatique, le paragraphe est, malgré tout, encore examiné. De cette façon, vous pouvez, par exemple, avec

```
\Clause{dummy}
```

sauter un paragraphe si le paragraphe pertinent a été supprimé dans une version ultérieure d'un contrat. Il convient de noter que la propriété de `dummy` comprend les valeurs nominales `vrai` et `faux`. Toutes les autres valeurs sont généralement ignorées et peuvent, dans le pire des cas, conduire à un message d'erreur.

```
\Clauseformat{nombre}
```

comme mentionné dans la déclaration précédente, les paragraphes juridiques et les `alinéas` sont habituellement numérotés. La mise en forme du nombre est faite en utilisant la commande `\Clauseformat` qui attend un seul argument, le nombre. Dans la prédéfinition comme :

```
\newcommand*{\Clauseformat}[1]{\S#1}
```

le nombre avec `\S` est seulement le symbole de `section` suivi d'une espace insécable préfixée, prêter attention à l'extension lors d'une redéfinition.

TABLE 10.3 – Caractéristiques possibles pour les arguments optionnels des commandes `\Clause` et `\SubClause`

---

<code>dummy</code>
le titre n'est pas défini mais décompté.
<code>head</code> =titre de colonne
titre de colonne est activé et utilisé dans ce titre de colonne quel que soit le titre du paragraphe.
<code>nohead</code>
aucun nouveau titre de colonne ne sera défini.
<code>notocentry</code>
aucune entrée n'est faite dans la table des matières.
<code>number</code> =numéro
le numéro des paragraphes est spécifié directement.
<code>preskip</code> =distance
nouvelle distance devant le titre du paragraphe.
<code>postskip</code> =distance
nouvelle distance après le titre du paragraphe.
<code>title</code> =titre
le paragraphe est prévu avec un numéro de titre. Ceci est également le réglage de base pour les éléments du titre de colonne et de la table des matières.
<code>tocentry</code> =liste de la table des matières
indépendamment d'un titre du paragraphe, cette entrée est faite dans la table des matières.

---

`juratitlepagebreak=on-off-valeur`

normalement, les sauts de page sont interdits dans les rubriques de toutes sortes. Cependant, quelques juristes ont besoin de pagination dans les titres de paragraphe, ce qui est autorisé par `juratitlepagebreak`, les réglages possibles `on,off,valeur` sont dans la table 1.6 page 20.

**clausemark=réglage**

Les paragraphes, division subordonnée avec une numérotation indépendante, ne génèrent aucun titre de colonne par défaut, mais des paramètres alternatifs peuvent cependant générer les titres. Voir l'option avec ses valeurs possibles et leurs significations dans la table 10.4 page suivante.

**10.3.6 Alinéas**

Dans les paragraphes de *scrjura*, les alinéas sont normalement numérotés automatiquement. Ils sont donc un élément fort de structure, semblable à `\paragraph` ou `\subparagraph` d'un texte normal. Dans les contrats, il est donc souvent apprécié de travailler avec un espace entre les paragraphes. Le paq de *scrjura* ne fournit pas, à cet effet, son propre mécanisme et pour cela, nous nous référons à l'option `parskip` les classes KOMA-Script (voir la section 3.10).

**parnumber=réglage**

La numérotation automatique des paragraphes correspond aux paramètres `parnumber=auto` et `parnumber=true`. Parfois, il est nécessaire de la désactiver, ce qui est possible avec `parnumber=false` et seul le numéro d'enregistrement est automatiquement réinitialisé. Pour mettre en œuvre cette option, il est nécessaire d'intervenir dans le mécanisme de paragraphe de L<sup>A</sup>T<sub>E</sub>X ce qui, dans certains cas rares, peut avoir un effet négatif et ce changement est alors à reprendre avec `parnumber=manuel`. C'est parfois l'inverse et, dans ce cas, il peut être réactivé avec `parnumber=auto`.

Les paragraphes se composant d'un alinéa unique sont automatiquement numérotés. Pour que cela fonctionne, deux paragraphes avec un chiffre identique ne peuvent pas être situés dans un même document. Puisque l'information est disponible seulement à la fin d'un paragraphe, il est généralement nécessaire de compiler deux fois jusqu'à ce que la numérotation automatique du paragraphe soit faite correctement.

```
par
\thepar
\parformat
\parformatseparation
```

La numérotation des paragraphes utilise le compteur `par` et sa présentation, `\thepar` est préconfigurée en chiffre arabe avec `\arabicpar`. Le numéro de paragraphe automatique est effectué avec `\parformat`. Ici `\thepar` est entre

TABLE 10.4 – Valeurs possibles de l’option `clausemark` pour générer des titres de colonnes par paragraphes

---

<code>both</code>
Les paragraphes produisent des marques gauche et droite pour l’exécution de titre de colonnes si le document d’utilisation fournit des titres en cours d’exécution.
<code>false, off, no</code>
Les paragraphes ne produisent pas de titre de colonne.
<code>forceboth</code>
Les paragraphes génèrent avec <code>\markboth</code> des marques gauche et droite pour titre de colonne indépendamment du fait que le document utilisé par exemple pour le style de page est toujours en cours d’exécution.
<code>forceright</code>
Les paragraphes génèrent avec <code>\markright</code> des marques à droite pour titre de colonne indépendamment du fait que le document utilisé par exemple sur le style de page est toujours en cours d’exécution.
<code>right</code>
Les paragraphes produisent des marques à droite pour titre de colonne si le document fournit l’utilisation des titres en cours d’exécution.

---

parenthèses. Un `paragraphe` avec un numéro préfixé manuellement doit l’être avec `\parformat` qui nécessite une espace insécable avec `\nobreakspace` ou l’utilisation du tilde.

Pour le nombre, `\parformat` est également utile pour ajouter un ou plusieurs délimiteurs qui sont définis par `\parformatseparation` et actuellement avec `\nobreakspace`, comme l’espacement de la mise en page des mots prédéfinis. Il convient de noter que `scrjura` suppose que `\thepar` est un chiffre arabe et que, par conséquent, cette déclaration ne devrait jamais être redéfinie.

<code>\ellipsispar[nombre]</code> <code>\parellipsis</code>
--

Parfois, dans les commentaires comparatifs, il est souhaitable d’omettre les paragraphes dans les lois, mais dans le même temps, de mettre l’omission en évidence. Les paragraphes doivent être pris en compte dans le comptage des paragraphes restants. Le paq `scrjura` utilise, dans ce cas, `\ellipsispar`.



Par défaut `\ellipsispar` peut oublier un, et même plusieurs paragraphes en utilisant le numéro d'argument optionnel. Dans tous les cas, un paragraphe non numéroté contenant uniquement les points de suspension apparaît dans la sortie `\parellipsis` en remplacement. Au moment de décider d'une numérotation automatique des paragraphes de la clause juridique, les paragraphes omis sont cependant inclus dans le décompte, ce qui vaut également pour les numéros de tous les paragraphes suivants.

Supposons que vous commentez le code pénal et voulez, au paragraphe 2, mettre seulement trois commentaires dans le paragraphe, tout en soulignant, néanmoins, les paragraphes restants.

Vous atteindrez votre but avec par exemple :

```
\documentclass[parskip=half]{scrartcl}
\usepackage{scrjura}
\usepackage{selinput}
\SelectInputMappings{
adieresis={ä},germandbls={ß},
}
\begin{document}
\begin{contract}
\Clause{title={Application dans le temps },number=2}
\ellipsispar[2]
Si l'on applique la loi sur la résiliation telle quelle est
modifiée avant la décision, alors une loi plus clémente sera
appliquée.
\ellipsispar[3]
\end{contract}
\end{document}
```

Pour voir le résultat, vous devriez juste essayer.

Les ellipses sont prédéfinies lorsque `\textellipsis` est utilisé pour définir une telle instruction. Sinon, `\dots` est utilisé. Une redéfinition des `\parellipsis` est possible à tout moment avec `\renewcommand`.

### 10.3.7 Archives

Les paragraphes dans les contrats consistent en un ou plusieurs ensembles, qui sont également numérotés en partie. Toutefois, étant donné qu'une numéro-

tation automatique est problématique et sujette à l'erreur, cette solution n'est pas mise en œuvre.

<code>sentence</code>
<code>\thesentence</code>
<code>\Sentence</code>

La numérotation des compositions se fait manuellement avec l'instruction `\Sentence`. Le compteur `composition` est automatiquement incrémenté et la sortie dans le cadre de `\thesentence` se fait par défaut en chiffres arabes. Avec `babel`, il est possible de définir facilement une abréviation de `\Sentence` :

```
\usesorthands'{}  
\defineshorthand'S{\Sentence\ignorespaces}
```

Cette définition est faite en sorte que les espaces soient ignorées par « S ». Il est même possible de définir une abréviation pour une période donnée, suivie d'un nouveau numéro d'enregistrement :

```
\defineshorthand'{}. {\Sentence\ignorespaces}
```

Ces deux abréviations ont fait leurs preuves dans la pratique même. Pour plus d'informations sur l'utilisation de `\usesorthands` et `\defineshorthand` se référer aux instructions du paq `babel` (Voir [BB13]).

### 10.3.8 Références croisées

Pour les paragraphes, les alinéas et les compositions, le mécanisme normal de références croisées avec `\label`, `\ref` et `\pageref` ne suffit plus. Par conséquent, cela est complété dans `scrjura` par d'autres instructions.

<code>\ref{étiquette}</code>
<code>\refL{étiquette}</code>
<code>\refS{étiquette}</code>
<code>\refN{étiquette}</code>

Les instructions `\refL`, `\refS` et `\refN` référencent complètement paragraphes, alinéas et compositions. Une longue forme textuelle utilise `\refL`, alors que `\refS` fournit un raccourci textuel et que `\refN` est un raccourci purement numérique. `\ref` correspond par défaut à `\refL`.

```
\refClause{étiquette}
\refClauseN{étiquette}
```

Pour le référencement de paragraphe, sans *alinéa* ni composition, `\refClause` et `\refClauseN` diffèrent seulement dans l'ancienne règle de fixation automatique du préfixe.

```
\refPar{étiquette}
\refParL{étiquette}
\refParS{étiquette}
\refParN[format numérique]{étiquette}
```

Seuls les paragraphes sont référencés avec `\refParL`, `\refParS` et `\refParN`.

Les formes diffèrent, suivant le cas, comme il est indiqué ci-dessus pour `\refL`, `\refN` et `\refS`. Une forme spéciale représente l'argument optionnel de `\refParN`.

Normalement les paragraphes sont référencés sous forme purement numérique en chiffres romains, mais un autre format peut être spécifié dans l'argument optionnel. Il est logique d'utiliser ici et en premier lieu, le format de chiffres arabes pour le numéro de paragraphe avec `\refPar` qui correspond par défaut à `\refParL`.

```
\refSentence{étiquette}
\refSentenceL{étiquette}
\refSentenceS{étiquette}
\refSentenceN{étiquette}
```

La composition sans paragraphe et l'*alinéa* peuvent être référencés avec `\refSentenceL`, `\refSentenceS` ou `\RefSentenceN`. Encore une fois, il existe une forme textuelle longue, une forme courte et une forme purement numérique, `\RefSentence` correspond par défaut au prépositionnement `\refSentenceL`.

```
ref=réglage
```

Les résultats de `\ref`, `\refPar` et `\refSentence` dépendent de l'option `ref` et correspondent par défaut à `\refl`, `\refParL` et `\refSentenceL`, les valeurs possibles et leur signification sont présentées dans la table 10.3 page 360.

Supposons que vous vouliez des paragraphes qui soient toujours référencés sous la forme de « paragraphe 1 alinéa 1 ». Il n'y a pas d'instruction prédéfinie à cet effet, malheureusement, et vous devez créer votre propre définition de cette possibilité d'aménagement. Vous pouvez le faire avec :

```
\newcommand*{\refParM}[1]{%
```

```
Paragraph~\refParN[arabic]{#1}  
dans Paragraph~\refClauseN{#1}%  
}
```

qui est très facile d'accès. La déclaration ainsi définie doit être utilisée comme `\refParL`. Des exemples de résultats non configurables avec des instructions de base figurent dans la table 10.2 page 322.

### 10.3.9 Environnements de contrats supplémentaires

Aucun utilisateur ne définit de contrat ou de commentaire individuel sur les lois avec `scrjura`, mais travaille dans différents domaines qui sont traités par des lois. Il est donc nécessaire que le paragraphe ne soit pas toujours fourni avec le même préfixe « §. » mais, par exemple, avec « Art. » ou encore avec « GG » (*Grundgesetz* – loi fondamentale allemande, prélude à la fusion de la RFA avec la RDA) ou toute autre définition qui sera appelée. En outre, un comptage indépendant des différents paragraphes est nécessaire.

L'instruction `\DeclareNewJuraEnvironment{nom}[options]{commencement}{fin d'instruction}` permet en outre de déclarer les environnements tels que textes contractuels ou légaux indépendants. L'argument deviendra le nom du nouvel environnement, les énoncés de début sont toujours exécutés au début de l'environnement, comme si vous les mettiez à chaque fois immédiatement après `\begin{nom}` et les énoncés de fin toujours exécutés à la fin, comme si vous les mettiez immédiatement avant `\end{nom}`, un environnement nouveau nommé avec ses propres compteurs est indiqué par une option vide, une liste d'options séparées par des virgules est possible.

Les options suivantes sont supportées :

- `Clause=instruction` indique sur quelle instruction dans l'environnement, la déclaration `\clause` doit agir. La déclaration attend un argument comme pour le contrat énoncé documenté. Pour une application correcte de cette option étendue, une connaissance de la fonction interne de `scrjura` est nécessaire.  
En outre, les exigences sur la déclaration d'une version peuvent encore changer. Par conséquent, il est actuellement recommandé de ne pas utiliser cette option.
- `SubClause=instruction`. donne accès à l'option `Clause`, avec `\SubClause` à la place de `\Clause`.
- `Sentence=instruction` indique l'environnement de l'instruction sur lequel la déclaration `\Sentence` doit être mappée. La `\declaration` ne

TABLE 10.5 – Valeurs possibles de l’option `ref` pour régler le format de `\ref`, `\refPar` et `\refSentence`

<code>long</code>	Les réglages <code>parlong</code> et <code>sentencelong</code> sont combinés.
<code>numeric</code>	Les paramètres <code>parnumeric</code> et <code>sentencenumeric</code> sont combinés.
<code>clauseonly</code> , <code>onlyclause</code> , <code>ClauseOnly</code> , <code>OnlyClause</code>	Les paramètres <code>paroff</code> et <code>sentenceoff</code> sont combinés. Ils caractérisent à la fois <code>\refPar</code> et <code>\refSentence</code> et produisent un résultat vide.
<code>parlong</code> , <code>longpar</code> , <code>ParL</code>	Le paragraphe est référencé sous une forme textuelle longue.
<code>parnumeric</code> , <code>numericpar</code> , <code>ParN</code>	Le paragraphe est référencé sous une forme purement numérique.
<code>paroff</code> , <code>nopar</code>	Le paragraphe n’est pas référencé. Il convient de noter que par <code>\refPar</code> crée un résultat vide
<code>parshort</code> , <code>shortpar</code> , <code>ParS</code>	Le paragraphe est référencé dans une forme courte.
<code>sentencelong</code> , <code>longsentence</code> , <code>SentenceL</code>	L’ensemble est référencé sous une forme textuelle longue.
<code>sentencenumeric</code> , <code>numericsentence</code> , <code>SentenceN</code>	L’ensemble est référencé sous une forme purement numérique.
<code>sentenceoff</code> , <code>nosentence</code>	La composition n’est pas référencée. Il convient de noter qu’une chaîne vide est générée, caractérisée par <code>\refSentence</code> .
<code>sentenceshort</code> , <code>shortsentence</code> , <code>SentenceS</code>	L’ensemble est référencé dans une forme textuelle raccourcie.
<code>short</code>	Les réglages <code>parshort</code> et <code>sentenceshort</code> sont combinés.

TABLE 10.6 – Exemple de tâches de l’option `ref` avec instructions indépendantes pour les références croisées

Déclaration	Exemple de sortie
<code>\refL{étiquette}</code>	§ 1 Paragraphe 1 Composition 1
<code>\refS{étiquette}</code>	§ 1 Par. 1 C. 1
<code>\refN{étiquette}</code>	§ 1 I 1.
<code>\refClause{étiquette}</code>	§ 1
<code>\refClauseN{étiquette}</code>	1
<code>\refParL{étiquette}</code>	Paragraphe 1
<code>\refParS{étiquette}</code>	Par. 1
<code>\refParN{étiquette}</code>	I
<code>\refParN[arabic]{étiquette}</code>	1
<code>\refParN[roman]{étiquette}</code>	i
<code>\refSentenceL{étiquette}</code>	Composition 1
<code>\refSentenceS{étiquette}</code>	C. 1
<code>\refSentenceN{étiquette}</code>	1.

devrait avoir aucun argument. Dans des circonstances normales, le compteur `sentence` avec `\refstepcounter` est incrémenté puis la sortie est faite sous une forme appropriée. Il est particulièrement nécessaire de faire en sorte qu’aucun espace indésirable ne soit installé.

- `ClauseNumberFormat=instruction` détermine la façon dont les numéros de paragraphe de cet environnement sont formatés pour la sortie. Une déclaration avec un seul argument est attendue : le nombre de paragraphes qui est le dernier argument d’une chaîne d’instructions, de sorte que cette chaîne peut également être ajoutée directement aux instructions.

Par exemple, dans l’introduction de cette section, définir l’environnement mentionné pour les articles avec :

```
\DeclareNewJuraEnvironment{article}[ClauseNumberFormat=Art.]
{}{}
```

est suffisant. Et si l’élément utilisant une classe `KOMA-Script` est réglé avec un espacement au lieu de l’indentation du paragraphe,

```
\DeclareNewJuraEnvironment{article}[ClauseNumberFormat=Art.
~]{\KOMAOptions{parskip}}{}
```

peut être utilisé et « Art. » est alors automatiquement préfixé lors du référencement en remplacement de « § ».

### 10.3.10 Prise en charge linguistique

Le paq *scrjura* a été développé en collaboration avec un avocat pour le droit allemand. Par conséquent, les langues `german`, `ngerman`, `austrian` et `naustrian` sont d'abord prises en charge. Le paq a été conçu de telle sorte qu'il fonctionne avec les langues habituelles de `babel`. Les utilisateurs peuvent facilement faire des ajustements avec `\providecaptionname` (voir section 12.4). Si vous avez des informations fiables sur le concept avec des termes d'environnement juridique d'autres langues, l'auteur apprécierait une rétroaction appropriée. De cette façon, le soutien pour `english`, `american`, `british`, `canadian`, `USenglish` et `UKenglish` a maintenant été ajouté.

TABLE 10.7 – Signification et préférences pour les termes dépendants de la langue qui ne sont pas encore définis

Déclaration	Signification	Par défaut
<code>\parname</code>	forme longue « paragraphe »	paragraphe
<code>\parshortname</code>	forme courte « paragraphe »	par.
<code>\sentencename</code>	forme longue « record »	ensemble
<code>\sentenceshortname</code>	forme raccourcie « set »	S.

```
\parname
\parshortname
\sentencename
\sentenceshortname
```

Les noms spécifiques à une langue sont utilisés par *scrjura*. La signification prédéfinie des valeurs figure dans la table 10.3 page 360. Les définitions de *scrjura* s'utilisent dans `\begin{document}` seulement si d'autres spécifications n'ont pas déjà été satisfaites. Si *scrjura* est utilisé avec une langue non prise en charge, un message d'erreur est émis.

### 10.3.11 Un exemple détaillé

Rappelons-nous le chapitre 4 dans lequel un membre de l'association a voulu conserver en mémoire la réunion en retard. Il devrait donc être possible qu'une prescription s'applique également à un club. Les statuts d'association sont en fin de compte une sorte de contrat qui peut être réglé avec `scrjura`

```
\documentclass[fontsize=12pt,\texttt{pagesize},parskip=half]
  {scrartcl}
```

lorsque la classe utilisée est `scrartcl`. Comme dans les lois courantes appliquées aux paragraphes, l'option `parskip=half` est utilisée (voir la section 3.10).

```
\usepackage[french]{babel}
```

Les statuts sont rédigés en français et par conséquent le paq `babel` option `french` chargé.

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage{textcomp}
```

Il introduit les paramètres par défaut de la police. En outre, le paq `textcomp` fournit en complément un symbole de l'euro utile pour certaines polices ainsi qu'une section également améliorée.

```
\usepackage{selinput}
\SelectInputMappings{
  adieresis={ä},
  germandbls={ß},
}
```

Les caractères spéciaux peuvent être saisis directement, l'encodage d'entrée est déterminé, sinon, le paq `inputenc` peut être utilisé ici aussi.

```
\usepackage{enumerate}
```

Modifier la numérotation des listes – par exemple remplacer les chiffres arabes par des lettres minuscules est possible avec le paq `enumerate`.

```
\usepackage[clausemark=forceboth,
  juratotoc,
  juratocnumberwidth=2.5em]
  {scrjura}
\useshorthands'{}
\defineshorthand'S{\Sentence\ignorespaces}
```



```
\defineshorthand'{}{. \Sentence\ignorespaces}
\pagestyle{myheadings}
```

Maintenant, il est temps de profiter de `scrjura`.

Avec l'option `clausemark=forceboth`, l'ensemble de marques gauche et droite pour les titres de colonnes (`Kolumnentitel`) dans les paragraphes est appliqué. En revanche, il n'est pas souhaitable que `\section` génère `Kolumnentitel` lorsque la page de style `myheadings` est utilisée, car normalement cette fonction n'est pas activée.

Ensuite, une table des matières pourra être créée et les paragraphes inclus, ce qui est réalisé avec `juratotoc`. Nous verrons plus loin que la largeur des numéros de paragraphe par défaut dans la table des matières ne suffit pas et par conséquent, comment paramétrer `juraticnumberwidth=2.5em` pour un peu plus d'espace.

Dans l'exemple, vous pouvez adopter des abréviations pour faciliter une saisie ultérieure.

```
\begin{document}
```

Il est temps de commencer un document réel :

```
\subject{Statuts}
\title{CAdBV}
\subtitle{Club des Amis du Bon Vin}
\date{01.\,04.\,2016}
\maketitle
```

Chaque document possède un titre de la loi qui est créé habituellement au moyen de `KOMA-Script` (voir section 3.7) :

```
\tableofcontents
```

Comme cela a déjà été mentionné, une table des matières doit être créée :

```
\addsec{Préambule}
```

Le club en France présente diverses facettes et, malheureusement, nous avons constaté que souvent, il souffre de sérieux face à la gravité du sujet. Les préambules ne sont pas rares dans les lois. Ici, il est défini par l'ordre `\addsec` pour recevoir une entrée dans la table des matières :

```
\appendix
```

Voici une petite astuce qui est utilisée. Les articles de la charte ne doivent pas être numérotés en chiffres arabes, mais avec des lettres majuscules - tout comme c'est aussi le cas dans l'annexe d'un article avec `scrartcl`.

```
\section{Généralités}
\begin{contract}
```

Et le contrat commence logiquement par le premier article :

```
\Clause{title={Nom, Statut juridique, Siège de l`association}}
L`~association porte le nom de Club des Amis du Bon Vin et `nest
pas inscrite dans un registre des associations.
L'association est un club inutile, non-économique. Elle n`~a pas
de siège et chacun doit donc se tenir debout.
L`~exercice se déroule, chaque année, du 31~Mars au 1~Avril.
```

Le premier paragraphe reçoit, en plus du numéro, un titre. Ce sera également ainsi dans les paragraphes suivants.

Le premier paragraphe du traité ne contient rien d'inhabituel. Comme il n'est pas le seul paragraphe, chaque paragraphe sera numéroté automatiquement. Pour que cela se produise aussi dans le premier paragraphe, deux compilations de L<sup>A</sup>T<sub>E</sub>X sont nécessaires, ce qui est déjà le cas pour la table des matières, mais cela ne perturbe pas la suite.

Le deuxième paragraphe comporte deux compositions. Ici, pour la première fois on trouve l'abréviation « C » et son application.

La première génère uniquement un numéro d'enregistrement.

La seconde génère non seulement un point, mais également un numéro pour l'enregistrement par la suite. Les deux sous-ensembles de ce paragraphe sont donc numérotés :

```
\Clause{title={But de l `association}}
\S Bien qu `elle n'a pas de sens, l'association n `est pas futile.
Rien n `est plus sérieux que de s `assurer de la qualité d `un bon
vin, sur une base saine, et avec gravité.
À cet effet, l'association avec :
- un verre dans l `nez;
- un casse-noix;
- un glaçon.
n'a aucun moyen \label{a: moyen}.
```

Le deuxième paragraphe contient plusieurs paragraphes qui comprennent, pour certains, également un ou des sous-ensembles. Dans le deuxième paragraphe, il

est possible aussi de trouver une liste et dans le dernier paragraphe, un ensemble d'étiquettes auquel on peut toujours se référer plus tard

```
\Clause{title={Bureaux de l 'association}
Les bureaux du club sont honorifiques.
\S La fonction CdA désigne le club avec ses moyens (voir
\ref {a:milieu}), afin qu'il puisse être géré à temps plein sans
opposer que ce n'est guère possible.
```

Le troisième paragraphe contient une référence croisée comme caractéristique spéciale, référencée dans ce cas dans la forme longue avec `clause`, paragraphe et sous-paragraphe. Si la composition est omise, alors cela pourra être réalisé plus tard en mettant l'option globale `ref=nosentence`.

```
\Clause{title={Club des Amis du Bon Vin},dummy}
\label{CdAdBV}
```

Il existe maintenant un premier cas particulier de paragraphe qui existait dans une version antérieure des statuts mais a ensuite été retiré. Les paragraphes suivants ne sont pas renumérotés, mais conservent leurs anciens numéros. Par conséquent, la déclaration `\Clause` n'a pas été supprimée, mais étendue à la propriété communautaire `dummy`. Elle permet de régler également une étiquette de ce paragraphe, bien que le paragraphe lui-même ne soit pas affiché :

```
\end{contract}
\section{Adhésion}
\begin{contract}
```

Dans l'article suivant, les problèmes avec la numérotation des paragraphes sont éliminés et l'environnement du contrat est arrêté momentanément :

```
\Clause{title={Types de membres},dummy}
```

En outre, le premier paragraphe de l'article suivant a été supprimé :

```
\Clause{title={Modalités d'adhésion}}
```

Chacun peut adhérer à un prix raisonnable auprès de l'un des `\refClause{CA-dBV}` et acquérir sa carte de membre dudit club :

```
\S-Pour être admissible à l'adhésion, un formulaire
d'inscription est requis. Cette demande doit être déposée,
écrite à l'encre verte sur papier rose.
```

Ici, nous avons à nouveau un vrai paragraphe qui fait référence à l'un des paragraphes supprimés. En second lieu, une étiquette est également définie :

`\SubClause{title={Ajout aux paragraphes ci-dessus}}`

`\S` Avec l'abolition de `\refClause {CdAdVB\}` `\ref{a: moyen\}` perd sa faisabilité et l'adhésion des membres peut alternativement être transmise.

Ici, nous avons le deuxième cas particulier d'un paragraphe, pas supprimé mais nouveau, inséré sans suivre la numérotation des paragraphes nouveaux avec l'utilisation de `\SubClause`. Ainsi, ce paragraphe prend le numéro de l'alinéa précédent, qui a été prolongé pour distinguer un petit « a » :

`\Clause{title={Fin de l'adhésion}}`

`\S` L'adhésion finit avec la vie. L'adhésion ne se termine pas lorsque les membres sont encore vivants.

`\Clause{title={Assemblée générale}}`

Deux fois par an, il y aura une assemblée générale.

Le délai entre deux assemblées générales ne doit pas dépasser six mois, 1 semaine et 2 jours. L'invitation à la prochaine réunion ne doit pas être faite plus tôt que 6 mois après la dernière réunion.

`\SubClause{title={Complément à l'~assemblée générale }}`

L'Assemblée générale doit se tenir au moins 2 semaines après la réception de la convocation.

`\end{contract}`

Les sections restantes de cet article ne contiennent que des instructions déjà connues et pas de nouvelles fonctionnalités :

`\section{Validité}`

`\begin{contract}`

`\Clause{title={Entrée en vigueur}}`

Cette Constitution prendra effet le 11 \11 \2011 à 11:11~ h moment où elle entre en vigueur.

`\S` Dans le cas où les dispositions du présent statut sont en contradiction avec les précédentes, le statut qui prenait effet le 11 \ 11 \ 2011 à 11: 11 .. ~ h et 11 secondes ~ est abrogé.

L'association est, dans ce cas, à considérer comme dissoute.

`\end{contract}`

Un autre article suit sans détail.  
`\end{document}`

## 10.4 Développement

Bien que le paq fasse partie de KOMA-Script depuis des années, utilisé par divers avocats, il a encore un numéro de version inférieur à 1. Par conséquent, il peut être considéré comme inachevé.

Il existe trois raisons à cela :

- Tout d’abord, le paq est conçu pour être utilisé comme généralement il l’est actuellement. Si par exemple il était prévu pour créer de prochains contrats dans d’autres environnements similaires, il apparaît que cette version couvre déjà les options de toute arrivée majeure. Cependant, il peut s’avérer utile de définir des nouveaux environnements de contrats, par exemple les articles de *La Loi fondamentale* en langue allemande, ce qui, maintenant, est chose faite.
- Ensuite, les fonctions testées avec toutes sortes d’environnements L<sup>A</sup>T<sub>E</sub>X, différentes classes et paqs, malgré une interaction constatée largement positive, me font considérer `scrjura` comme un paq très spécial, bien au-delà de la pratique quotidienne de l’auteur qui est, à cet égard, dépendant dans une large mesure des commentaires éclairés des utilisateurs.
- Enfin, les choses dans `scrjura` peuvent naturellement évoluer en dérivant naturellement, du moins je l’espère, du plus petit numéro de version. Bien que l’auteur travaille en ce sens pour préserver la compatibilité avec les versions antérieures, cela ne peut cependant pas être, à ce jour, garanti.



# Chapitre 11

## KOMA-Script pour utilisateurs avancés et experts

Dans cette partie sont regroupées des informations sur les classes  $\text{\LaTeX}$  à destination des créateurs de paqs. Elles précisent non seulement les instructions utiles pour la mise en œuvre des paqs et classes, mais aussi les interfaces nécessaires à la création avec **KOMA-Script**. Elle comporte, en outre, des précisions sur les options obsolètes ainsi que des notes de fond sur l’optimisation de **KOMA-Script**. Le chapitre II est destiné à compléter les informations à destination des auteurs d’articles, de rapports, de livres et de lettres, avec plus de notes et davantage d’exemples pour les utilisateurs.

### 11.1 Fonctions de base de `scrbase`

Le paq `scrbase` fournit des fonctionnalités de base conçues et mises en œuvre à l’usage des auteurs de paqs et de classes. Il peut aussi être utilisé pour les classes `Wrapper` liées ou non à la classe **KOMA-Script**. Ainsi les auteurs de classes étrangères à **KOMA-Script** bénéficient de toutes les fonctionnalités de `scrbase`.

## 11.2 Charger le paq

Un utilisateur charge un paq en employant `\usepackage`, un auteur préférera `\RequirePackage`.

Les paqs tels que `Wrapper` nécessitent `\RequirePackageWithOptions` mais la commande `\RequirePackage` utilise les mêmes arguments optionnels que `\usepackage`.

En revanche, `\RequirePackageWithOptions` n'a pas un argument facultatif et passe toutes les options proposées lors du chargement du paq `Wrapper` pour le paq requis. Voir [Tea06] pour plus d'informations sur ces commandes.

Le paq `scrbase` nécessite, en interne, l'utilisation des fonctionnalités du paq `keyval` qui peuvent être fournies alternativement par le paq `xkeyval`. Le paq `keyval` gère les définitions de clés et les attributions de valeurs à ces clés. Il est possible d'utiliser, avec `scrbase`, la syntaxe `keyval : key=value` (clé=valeur).

`\internalonly=valeur`

Le paq `scrbase` fournit pour une exécution conditionnelle des commandes internes dont les noms primaires ressemblent à `\scr@name`, utilisées également et uniquement en interne par `KOMA-Script`. L'auteur de paqs et de classes peut utiliser ces commandes mais ne devrait pas les redéfinir car d'autres paqs peuvent comporter des ordres avec le même nom mais une syntaxe ou des fonctionnalités différentes. Pour éviter un conflit, `scrbase` peut en modifier le nom et la définition. L'option `internalOnly` permet de modifier l'exécution conditionnelle des commandes internes en entrant une option sans valeur qui définit les instructions de branchement internes.

Alternativement, l'utilisateur précise les commandes qui ne doivent pas être définies comme valeur, en donnant la priorité au remplacement de « `\` » par « `/` ».

En principe, les auteurs de paqs et de classes ne devraient pas faire usage de cette option. L'utilisateur peut spécifier avec ou sans valeur, soit comme option globale dans `\documentclass` soit via `\PassOptionsToPackage`.

Si l'utilisateur, par exemple, ne veut pas que `scrbase` définisse les commandes `\ifVtex` et `\ifundefinedorrelax`, il charge la classe :

```
\documentclass%
    [internalonly=/ifVtex/ifundefinedorrelax]%
    {foo}
```



Le nom de classe `foo` est utilisé ici comme un espace réservé.

Pour la signification des commandes `\ifVtex`, `\ifundefinedorrelax` et autres instructions conditionnelles, reportez-vous à la section 11.3.

## 11.3 Caractéristiques des familles et extensions

Comme déjà mentionné, `scrbase` utilise le paq `keyval` pour les clés et leurs valeurs, il en étend la fonctionnalité à une seule famille qui détient toutes les clés et dont il reconnaît tous les membres. Par conséquent, une clé peut être détenue par une famille ou par un ou plusieurs de ses membres et une valeur attribuée à la clé d'une famille, d'un membre de la famille ou de tous les membres de la famille.

```
\DefineFamily{nom de famille}
\DefineFamilyMember[membre]{nom de famille}
```

Pour diverses raisons `scrbase` doit connaître les membres de la famille. Par conséquent, il est nécessaire de définir une nouvelle famille avec `\DefineFamily` qui crée une liste de membres vide. Si la famille existe déjà, il ne se passera rien, ce qui signifie qu'une liste de membres déjà existante ne serait pas remplacée. Ensuite, un nouveau membre peut être ajouté à l'aide de `\DefineFamilyMember`. Si le membre existe déjà, rien ne se passe, mais l'ajouter à une famille qui n'existe pas provoque un message d'erreur. Si aucun membre n'est spécifié, la valeur par défaut `\@currname\@currext` est utilisée. Lors du chargement d'une classe ou du paq `\@currname` et `\@currext` composent ensemble le nom de fichier de la classe ou du paq.

Théoriquement, il est possible de définir un membre sans nom à l'aide d'un argument optionnel vide de membre de la famille qui correspondrait à la famille elle-même. Il est recommandé de n'utiliser que des lettres et des chiffres pour la famille mais surtout que le premier caractère d'un membre de la famille ne soit ni une lettre, ni un chiffre pour éviter que les membres d'une famille soient les mêmes que les membres d'une autre famille. `scrbase` s'attribue l'appartenance à la famille « KOMA » et lui ajoute son fichier `scrbase.sty` comme membre. En principe, les noms de famille « KOMA » et `KOMA-Script` sont réservés. Pour vos propres paqs, il est recommandé d'utiliser le nom global en tant que nom de la famille et le nom des paqs individuels en tant que membres.

Supposons que vous écrivez un nouveau paq `boucher` qui contient les paqs `salami.sty`, `merguez.sty` et `boudin.sty`.

Vous décidez d'utiliser le nom de famille `boucher` et, pour chacun des fichiers de paqs, vous ajoutez les lignes :

```
\DefineFamily{boucher}
\DefineFamilyMember{boucher}
\DefineFamilyMember{salami}
\DefineFamilyMember{merguez}
\DefineFamilyMember{boudin}
```

Lors du chargement des trois paqs, cela va ajouter les membres `salami.sty`, `merguez.sty`, et `boudin.sty` à la famille `boucher`. Après les avoir chargés, les trois membres sont alors définis.

```
\DefineFamilyKey[membre]{famille}{key} [valeur par défaut]{action}
\FamilyKeyState
\FamilyKeyStateUnknown
\FamilyKeyStateProcessed
\FamilyKeyStateUnknownValue
\FamilyKeyStateNeedValue
```

La commande `\DefineFamilyKey` définit une clé. Si un membre de la famille est donné, la clé devient un attribut de ce membre. Si aucun membre n'est indiqué, le membre `\@currname\@currext` est donné par défaut. Si, plus tard, une valeur est affectée à la clé, l'action est exécutée avec la valeur qui devient argument d'action interne « # 1 » et prend cette valeur. Une valeur omise amène l'utilisation de la valeur par défaut. S'il n'existe pas de valeur par défaut, la clé ne peut être utilisée qu'avec une valeur définie. Cela conduit finalement à :

```
\DefineFamilyKey[membre]{famille}{key}
[valeur par défaut]{action}
```

à l'appel de :

```
\définir@key{membre de la famille}{key}
[valeur par défaut]{action prolongée}
```

où `\define@key` est déterminée dans le paq `keyval` (voir [Car99a]). Cependant, répondre à l'appel de `\define@key` nécessite quelques précautions supplémentaires et l'action est élargie pour inclure ces mesures.

Le succès ou l'échec d'exécution de l'action sera présenté par `\FamilyKeyState` à `scrbase` afin de prendre des mesures supplémentaires si nécessaire et utiliser l'option par défaut `\FamilyKeyState` identique à `\FamilyKeyStateUnknown`. Cela signifie que l'on ne sait pas si l'exécution s'est faite correctement et si l'état ne change pas jusqu'à la fin de l'action, `scrbase` écrira une note dans le

fichier journal reprise par `\FamilyKeyStateProcessed` accepté dans la procédure ultérieure.

L'état `\FamilyKeyStateProcessed` indique que l'option et l'affectation de la valeur a été achevée et que tout est en ordre. Cette condition peut être facilement changée en appelant `\FamilyKeyStateProcessed`.

L'état `\FamilyKeyStateUnknownValue` indique que l'option a été traitée, mais qu'une valeur doit être affectée, ce qui n'est pas autorisé. Cet état, par exemple, rapporte qu'il essaye la possibilité d'affecter à `twoside` une valeur inconnue avec `typearea`. Le choix de l'état se fait simplement en appelant `\FamilyKeyStateUnknownValue`.

L'état `\FamilyKeyStateNeedValue` indique que l'option ne peut être traitée car une valeur est nécessairement attendue et l'appel a été fait sans affectation de valeur. Cet état est défini spontanément quand une option échoue car elle est utilisée sans affectation de valeur, En appelant `\FamilyKeyStateNeedValue` un changement est théoriquement possible. En outre, des messages d'erreurs supplémentaires peuvent être définis à l'aide de `\FamilyKeyState`. En général, les quatre états prédéfinis suffisent et devront donc être employés.

Supposons que chacun de nos trois paqs obtienne une clé nommée `charcuterie`. Lorsqu'il est utilisé, un commutateur doit être fixé à chaque paq en conséquence avec la valeur de l'appel. Le paq `salami` pourrait ressembler à ceci :

```
\newif\if@Salami@charcuterie
\DefineFamilyKey{boucher}%
{charcuterie}[true]{%
\expandafter\let\expandafter\if@Salami@charcuterie
\csname if#1\endcsname
\FamilyKeyStateProcessed
}
```

La valeur est donc `true` ou `false` lors de l'appel autorisé. Un test pour des valeurs non valides n'existe pas dans cet exemple. Au lieu de cela, il est toujours fait rapport que l'option a été traitée complètement et correctement. La clé ne sera utilisée que plus tard si l'une des valeurs autorisées doit être affectée à un appel ou sans une affectation de valeur. Dans ce dernier cas la valeur `true` est attribuée par défaut.

Les deux autres paqs peuvent être définis d'une manière identique et dans ce cas, seul « salami » doit être remplacé par les noms correspondants.

```
\RelaxFamilyKey[membre]{famille}{clé}
```

La clé est définie comme attribut d'un membre de la famille, mais la définition est pratiquement éliminée, la clé ne définit plus ce membre et l'utilisation d'une clé qui n'est pas définie pour ce membre de la famille est également autorisée.

```
\FamilyProcessOptions[membre]{famille}
```

En principe, l'extension des familles de clés aux familles et aux membres est prévue et la clé ou l'affectation de sa valeur utilisée comme option de classe ou de paq. Cette déclaration représente une expansion de l'option `\Process*` à partir du noyau de L<sup>A</sup>T<sub>E</sub>X, (voir [Tea06]), qui traite l'instruction non seulement des options qui ont été définies par `\DeclareOption`, mais aussi les clés d'un membre spécifié de la famille traitée. Si l'argument optionnel `membre` n'est pas spécifié, l'option `membre \@current.\@currname` sera utilisée.

Les clés de recherche de fonctions spéciales sont définies avant les clés des membres de la famille lorsqu'elles ne sont pas associées à un membre de la famille, ou sont associées à une famille dans chaque membre est resté vide.

Si dans l'exemple précédent, un paq est prolongé par la ligne :

```
\FamilyProcessOptions{boucher}
```

l'utilisateur peut charger le paq et sélectionner les propriétés des viandes. Avec une option globale, ce qui est indiqué dans l'argument optionnel `\documentclass`, la propriété est automatiquement réglée pour les trois paqs lorsque les trois paqs sont chargés. Les paqs traitent toujours les options globales avant les options locales. Lors du traitement des options globales, une valeur d'option inconnue est inscrite dans le fichier journal, ce qui entraîne un message d'erreur, et provoque le renvoi de l'option globale vers les options locales.

Interpréter `\FamilyProcessOptions` comme une extension de `\ProcessOption*` ou une extension de `keyval clé=valeur` du mécanisme, en fin de compte, avec l'aide de `\FamilyProcessOptions, clé=valeur` devient une option.

Lors de l'exécution des options, `\FamilyProcessOptions` ne doit pas être utilisé. Il est donc interdit dans l'exécution des options pour charger les paqs qui auto-utilisent `\FamilyProcessOptions`.

```
\BeforeFamilyProcessOptions[membres]{famille}{code}
```

En particulier, les auteurs de classes `Wrapper` souhaitent parfois intervenir dans un paq ou une classe avant de traiter les options d'un futur paq ou d'une classe à venir chargés avec `\FamilyProcessOptions`.

Ceci est possible avec `\BeforeFamilyProcessOptions`.

Le paq `scrbase` prévoit pour cela les crochets qui peuvent être ajoutés avec la commande `\BeforeFamilyProcessOptions`. Les paramètres des membres et de la famille correspondent à ceux de `\FamilyProcessOptions`. Cependant, vous pouvez également utiliser, entre crochets, les membres de la famille. Utiliser ce code si la famille ou la personne n'a pas encore été définie.

Le crochet d'un membre de la famille est de toute façon supprimé automatiquement après son exécution. Toutefois, si un membre reste vide, ce crochet est exécuté pour tous les membres de la famille et restera maintenu pour une mise en œuvre ultérieure.

Vous écrivez un paq qui charge le paq `saucisse` et vous ne voulez pas que, pour ce paq, l'option de coupe soit définie. Par conséquent, vous désactivez cette option avant de charger le paq sur `\BeforeFamilyProcessOptions` :

```
\RequirePackage{scrbase}
\BeforeFamilyProcessOptions[saucisses.sty]{boucher}{%
\RelaxFamilyKey[saucisses.sty]{boucher}{charcuterie}%
}
\RequirePackageWithOptions{saucisses}
```

Maintenant, quelqu'un veut essayer votre paq avec l'option pour charger des coupes, et se connecte avec le paq `saucisses`, bien que cette option ne soit pas connue. Si l'option `charcuterie` est spécifiée comme une option globale, le paq `saucisses` est ignoré. Les préférences internes du paq, par exemple avec `\FamilyExecuteOptions` avant `\FamilyProcessOptions` sont cependant indépendantes.

```
\FamilyExecuteOptions[membre]{famille}{liste d'options}
```

Cette commande est une extension de `\ExecuteOptions` du noyau  $\LaTeX$  (voir [Tea06]). Les processus de commande traitent non seulement les options qui sont définies à l'aide de `\DeclareOption`, mais aussi tous les processus clés des membres d'une famille donnée. Si l'argument optionnel `membre` n'est pas spécifié, alors `\@Currname \@Currext` est utilisé.

Une singularité : les clés qui ne sont pas associées à un membre de la famille, mais à une famille dont chaque membre est resté vide sont définies comme clés des membres de la famille.

Si l'option `charcuterie` doit être définie dans l'exemple précédent, les paqs n'utilisent qu'une seule ligne :

```
\FamilyExecuteOptions{boucher}{charcuterie}
```

qui sera ajoutée.

Cette déclaration peut également être utilisée dans d'autres options d'exécution.

```
\FamilyOptions{famille}{liste des options}
```

La liste des options prend la forme :

clé=valeur, clé=valeur...

où quand les clés présentent une carence définie de valeur, l'attribution de la valeur peut également être omise.

Contrairement aux options régulières définies par `\DeclareOption`, la clé peut être réglée après le chargement d'une classe ou du paq. À cet effet, l'utilisateur use de `\FamilyOptions`. Les clés spécifiées de tous les membres de la famille y sont accessibles. S'il existe une clé comme attribut de la famille elle-même, elle est placée en premier. Après cela, les clés de membre suivent dans l'ordre dans lequel les éléments ont été définis. Si une clé donnée n'existe pas, que ce soit pour la famille ou pour un membre de la famille, le paq `scrbase` signale une erreur. Cela peut se produire, bien que certains membres possèdent une clé, mais chacun de ces membres signale une erreur sur `\FamilyKeyState`.

Vous étendez votre projet `boucherie` avec `salade de saucisse`. Si le progiciel est utilisé, tous les paqs doivent d'abord produire la charcuterie :

```
\Providespack{salade de saucisse}%
[2008/05/06 nonsense package]
\RequirePackage{scrbase}
\DefineFamily{boucher}
\DefineFamilyMember{boucher}
\FamilyProcessOptions{boucher} \relax
\FamilyOptions{boucher}{charcuterie}
```

Si aucun paq `saucisse` n'est chargé, un message d'erreur sera émis en raison de l'option `undefined` de `charcuterie`. Ceci peut être évité par une clé appropriée définie avant la dernière ligne pour le paq lui-même :

```
\DefineFamilyKey{boucher}%
           {charcuterie}[true]{}%
```

produisant ainsi des paqs `saucisse non coupée` qui sont chargés après `salade de saucisse`. Cela peut aussi changer pour :

```
\AtBeginDocument{
\DefineFamilyKey{. salade de saucisse;styl}%
                {boucher}\%
```

```

                                {charcuterie}[true]{}%
}
\DefineFamilyKey{boucher}%
                                {charcuterie}[true]{}%
\AtBeginDocument{\FamilyOptions{boucher}{charcuterie}{%
}%
}%

```

Ainsi, la première option définie avec `\begin{document}` n'a plus de fonction pour le paq `salade de saucisse`, à ce moment, `\@currname` et `\@current` ne sont plus valables et l'argument optionnel de `\DefineFamilykey` doit être utilisé. Jusqu'à sa redéfinition, une définition est utilisée par défaut, tandis que le `\begin{document}` exécute à nouveau l'option pour toute la famille et l'établit aussi à d'autres paqs `saucisse`.

`\FamilyOption{famille}{option}{liste de valeurs}`

En plus des options qui ont des valeurs mutuellement exclusives, il peut y avoir des options avec plusieurs valeurs en même temps. Pour cela, il serait nécessaire lors de l'utilisation de `\FamilyOptions`, d'assigner à plusieurs reprises, une valeur différente à l'option et ainsi de préciser à plusieurs reprises les options. Vous pouvez aussi attribuer, à une option unique de `\FamilyOption`, toute une liste de valeurs.

La liste des valeurs est composée d'éléments séparés par des virgules :  
 valeur, valeur...

Dans ce contexte, l'utilisation d'une virgule n'est possible que si les valeurs sont placées entre accolades. La fonctionnalité générale de cette commande est la même que `\FamilyOptions`.

Exemple : le paq `salade de saucisse` doit contenir une option par laquelle vous pouvez déterminer d'autres ingrédients. Un commutateur est défini pour chaque ingrédient :

```

\newif\if@salade avec@oignon
\newif\if@salade avec@concombre
\newif\if@salade avec@poivron
\DefineFamilyKey{boucher}{salade supplémentaire}{%
    \csname@salatmit@#1true\endcsname
}

```

Ici les trois ingrédients « oignon », « concombre » et « poivron » ont été définis. Un message d'erreur pour les ingrédients « non définis » n'existe pas.

Pour une salade avec oignons et concombres, l'utilisateur peut employer :

```
\FamilyOptions{boucher}{%
{Salade supplémentaire=salade, oignon, concombre addendum},
```

ou plus simplement :

```
\FamilyOptions{boucher}{%
{ingrédients salade}{oignon,cornichon}
```

ou encore utiliser :

```
{ingredient=oignon,ingredient=cornichon}
```

```
\AtEndOfFamilyOptions{action}
```

Parfois, il est utile de retarder l'exécution d'une action qui s'inscrit dans le cadre de l'affectation de valeur à une clé jusqu'à ce que tous les travaux internes de `\FamilyProcessOptions`, `\FamilyExecuteOptions`, `\FamilyOptions` ou de `\FamilyOption` soient terminés. Ceci peut être effectué à l'aide d'une définition de `\AtEndOfFamilyOptions`. Le rapport d'état de défaillance de l'action n'est pas possible dans ce cas. En outre, la commande ne doit pas être utilisée en dehors d'une définition de l'option.

```
\FamilyBoolKey[membre]{famille}{clé}{nom de l'interrupteur}
\FamilySetBool{famille}{clé}{nom de l'interrupteur}{valeur}
```

Dans les exemples précédents les commutateurs sont utilisés plusieurs fois. Dans l'exemple avec l'option `charcuterie`, il est nécessaire d'indiquer une valeur de l'utilisateur telle que `true` ou `false`. Il n'y a pas de message d'erreur si l'utilisateur fait usage d'une valeur erronée. Les interrupteurs booléens sont une caractéristique d'application commune qui permet de paramétrer avec facilité, dans `scrbase`, `\FamilyBoolKey` pour la définition des options de recherche. Les arguments de la famille, du membre de la famille et de la clé sont les mêmes que ceux utilisés par `\DefineFamilyKey`.

Le nom de l'argument du commutateur est celui de ce commutateur sans le préfixe `\if`, si un commutateur de ce nom n'existe pas déjà. Il est défini et initialisé automatiquement à `false` par `\FamilyBoolKey` qui remplace en interne `\FamilySetBool` par `\DefineFamilyKey`. La valeur par défaut pour une telle option est toujours `true`. D'autre part, `\FamilySetBool` comprend l'ajout des valeurs `on` et `off` pour allumer et éteindre le commutateur. Si une valeur inconnue est passée, la commande `\FamilyUnkownKeyValue` est appelée avec la



famille des arguments et la clé et la valeur sont reprises dans `\FamilyKeyState`. Un message d'attribution de valeur inconnue est affiché si nécessaire.

La clé `charcuterie` doit être définie de manière un peu plus explicite dans les paqs `saucisse` qui utilisent, en outre, la même clé de sorte que tous les paqs, ou aucun ne produira le mot `charcuterie` :

```
\FamilyBoolKey{boucher}{charcuterie}%
                                {@charcuterie}
```

Un test avec « `charcuterie` » devrait ressembler à ceci :

```
\if @charcuterie ...
\else ...
\fi
```

Ce serait alors identique dans les trois paqs `saucisse`. Ainsi, on pourrait définir la propriété `charcuterie` en tant que propriété de principe de la famille :

```
@ifundefined{if@charcuterie}{%
\expandafter\newif\csname if@charcuterie\endcsname
}{}%
\DefineFamilyKey[] {boucher}{charcuterie} [true] {%
\FamilySetBool{boucher}{charcuterie}%
{@charcuterie}%
{#1\%
}
```

ou plus simplement :

```
\FamilyBoolKey[] {boucher}{charcuterie}%
                                {@charcuterie}
```

profitant de la référence secondaire qui s'applique à `\DefineFamilyKey`, mais aussi à `\FamilyBoolKey`.

Puisque `\FamilyKeyState` est déjà défini par `\FamilySetBool`, la définition de l'option peut être aussi demandé avec `\DefineFamilyKey` et vous pourriez dans le premier cas, par exemple, après `\FamilySetBool` réaliser un test du type :

```
\ifx\FamilyKeyState\FamilyKeyStateProcessed
...
\else...
\fi
```

l'utilisation d'informations supplémentaires est valable pour `\DefineFamilyKey`, `\FamilyBoolKey` et pour `\FamilySetBool`. À ce stade, selon l'exécution de `\FamilySetBool` réussie ou non, un test utilisant `\ifx` est nécessairement fait. Les essais d'expansion tels que `\ifstr` sont à éviter ici.

```
\FamilyNumericalKey[membre]{famille}{clé}[valeur d'échec] {nom de
macro}{liste de valeurs}
\FamilySetNumerical{famille}{clé} {nom de macro}{liste de
valeurs}{valeur}
```

Tandis que les commutateurs ne prennent que deux valeurs, une clé admet des valeurs multiples. Par exemple, un alignement peut être à gauche ou ne pas l'être, mais aussi à gauche, au centre ou à droite. En interne, cette déclaration de  $\TeX$  est paramétrées en utilisant `\ifcase` et attend une valeur numérique. Par conséquent, dans `scrbase`, on peut attribuer une définition de macro via une clé en utilisant `\FamilyNumericalKey`.

La liste des valeurs a la forme :

```
{valeur}{definition},{valeur}{definition},...
```

La liste de valeurs ne définit pas uniquement les valeurs autorisées et prises en charge par la clé, mais chacune de ces valeurs si la définition de commande `\macro` est donnée. Habituellement il s'agit d'une valeur numérique.

Néanmoins, d'autres contenus sont possibles et permis. Il existe actuellement une restriction, mais cette définition doit être entièrement extensible et doit se développer dans l'attribution en interne.

La « saucisse » de « salade de saucisse » peut être coupée de diverses façons. Il est concevable que la saucisse soit coupée en lanières grossières, en lanières plus fines, ou reste non coupée. Cette information doit être stockée dans la déclaration `\coupe` de salade :

```
\FamilyNumericalKey{boucher}%
{section salade}{section}{%
{aucune}{rien},{non}{point},%
{grossier}{grossier},%
{fin}{fin}%
}
```

Dans ce cas, aucune coupe par l'utilisateur, ni avec :

```
\FamilyOptions {boucher}{salade coupe=aucune}
```

ou ni même avec :

```
\FamilyOptions {boucher}{salade coupe=point}
```

Dans les trois cas `\coupe` ne serait pas définie avec un contenu. Il peut être utile, comme dans cet exemple, de proposer des valeurs multiples à l'utilisateur pour le même résultat. En outre, il est probable que « coupe » ne soit pas éditée, mais évaluée plus tard. Dans ce cas, une définition textuelle ne serait pas pratique, mais si l'on définit la clé ainsi, elle pourra être utilisée par :

```
\FamilyNumericalKey{boucher}%  
{section de salade}{section}{%  
{aucune}{0},{non}{0},{point}{0}%  
{gros}{1},%  
{fin}{2}%  
}
```

et vous pourrez ensuite distinguer facilement une condition semblable qui sera utilisée sous la forme :

```
\ifcase\coupe  
    \% pas de coupe  
\or  
    \% grossièrement haché  
\else  
    \% haché finement  
\fi
```

Lors de l'appel, `\FamilyKeyState` interprète automatiquement la réussite ou l'échec.

En interne de `\FamilyNumericalKey`, `\DefineFamilyKey` est alors utilisé avec l'instruction `\FamilySetNumerical`. Si une valeur inconnue est assignée à une clé, `\FamilySetNumerical` appelle `\FamilyUnkownKeyValue` avec arguments de la famille, clé et valeur.

Cela conduit à une signalisation d'erreur dans `\FamilyKeyState` utilisant la condition `\FamilyKeyStateUnkownKeyValue` et, par exemple, lorsqu'il est utilisé comme une option locale à un message d'erreur. De même, lorsque vous appelez `\FamilySetNumerical` le succès via `\FamilyKeyStateProcessed` est signalé dans `\FamilyKeyState`.

```
\FamilyCounterKey[membre]{famille}{clé}[valeur par défaut]
{\LaTeX-compteur}
\FamilySetCounter{famille}{clé}{\LaTeX-compteur}{valeur}
```

Alors que dans `\FamilyNumericalKey` une macro est réglée sur une valeur numérique basée sur une valeur symbolique correspondante, il existe des cas dans lesquels une clé représente directement un compteur de `\LaTeX` avec une valeur numérique à attribuer immédiatement. C'est le but de la commande interne `\FamilyCounterKey` puis de `\FamilySetCounter` qui est appelée.

Des contrôles de valeur de l'argument de base sont effectués pour déterminer si cet argument convient pour une affectation à ce compteur.

L'attribution n'est faite que si les tests réussissent. Les erreurs détectées sont signalées par `FamilyKeyStateUnknownValue`.

Cependant, il est possible que toutes les erreurs ne soient pas détectées de sorte qu'une allocation incorrecte peut aussi conduire à un message d'erreur de `\TeX` lui-même.

Si aucune valeur n'est passée, la valeur par défaut est utilisée. Si aucune valeur par défaut n'est spécifiée, la clé peut être utilisée plus tard uniquement avec le transfert d'une valeur.

```
\FamilyCounterMacroKey[membre]{famille}
{clé}[valeur par défaut]{macro}
\FamilySetCounter{FamilySetCounterMacro}
{famille}{clé}{macro}{valeur}
```

Ces deux déclarations sont différentes de `\FamilyCounterKey` précédemment déclarée et `\FamilySetCounter` seulement par le fait qu'aucun compteur de `\LaTeX` n'est réglé à une valeur, mais qu'une macro est définie par cette valeur.

```
\FamilyLengthKey[membre]{famille}{clé}[valeur par
défaut]{longueur}
\FamilySetLength{clé}{longueur}{valeur}
\FamilyLengthMacroKey[membre]{famille}{clé}[valeur par
défaut]{macro}
\FamilySetLengthMacro{famille}{clé}{macro}{valeur}
```

À propos de `\FamilyLengthKey` une clé peut être définie représentant une longueur. Il importe peu de savoir s'il s'agit de longueur `\LaTeX`, de distance ou d'extension `\TeX` qui est utilisée. En interne, la longueur est définie par `\FamilySetLength`. Mieux vaut vérifier l'argument de la valeur de base plutôt que de déterminer s'il est adapté pour une affectation à une longueur.

La répartition n'a lieu que si les tests réussissent. Cependant, tous les défauts ne sont pas détectés et une attribution incorrecte peut conduire à un message d'erreur T<sub>E</sub>X. Les erreurs sont signalées par `\FamilyKeyStateUnknownValue`.

Si aucune valeur n'est passée, la valeur par défaut est utilisée à la place. Si la valeur par défaut manque, la clé peut être utilisée plus tard uniquement avec un transfert de valeur.

Avec les commandes `\FamilyLengthMacroKey` et `\FamilySetLengthMacro` le stockage de la valeur n'a pas lieu dans une `\longueur`, mais dans une `\macro`.

```
\FamilyStringKey[membre]{famille}{clé}[valeur par défaut]{macro}
```

Ceci définit une clé qui peut prendre n'importe quelle valeur. La valeur sera stockée dans la macro spécifiée et s'il n'existe aucun argument optionnel pour la valeur par défaut, le code `\FamilyStringKey` correspond à :

```
\DefineFamilyKey[membre]{famille}{clé}
      {\defMakro {#1}}
```

Elle indique, en outre, si l'argument optionnel pour la valeur par défaut est omis avec l'appel de :

```
\DefineFamilyKey[membre]{famille}{clé}
      [valeur par défaut]
      {\defMakro {#1}\FamilyKeyStateProcessed}.
```

Si la commande n'est pas définie, elle est définie comme une macro vide.

Si par défaut, 250 g de salade de saucisse sont produits, ce montant doit être simple à modifier à l'aide d'une option. À cet effet, la quantité doit être créée par la macro enregistrée `\poids` de la salade. L'option qui change le poids doit pouvoir également gérer le poids de la salade chaude :

```
\newcommand* {\poidsdesalade}{250g}
\FamilyStringKey {boucher}%
{poidsdesalade}[250g]{\poidsdesalade}
```

Pour revenir au poids par défaut après l'avoir modifié, l'utilisateur peut employer simplement l'option sans poids :

```
\FamilyOptions{boucher}{poidsdesalade}
```

Ceci est possible parce que la quantité standard a été fixée dans la définition comme valeur par défaut. Dans ce cas, il n'y a pas de valeur inconnue car toutes les valeurs sont simplement utilisées pour une définition de macro. Cependant, dans cette attribution, aucune valeur ne peut être incluse dans la clé.

<code>\FamilyUnknownKeyValue{<i>famille</i>}{<i>clé</i>}{<i>valeur</i>}{<i>liste des valeurs</i>}</code>
--

Une valeur inconnue ou non autorisée provoque un message d'erreur signalé par `\FamilyKeyState`. La liste de valeurs autorisées, séparées par une virgule, se présente sous la forme :

`' valeur ', ' valeur '...`

Actuellement, les valeurs ne sont pas évaluées par la liste de `scrbase`.

Nous réglons d'abord, à l'aide de `\FamilySetBool`, le commutateur booléen pour les coupes, nous testons ensuite si `\FamilyKeyState` signale la réussite de la commande à l'état `\FamilyKeyStateProcessed`. Si c'est un succès la coupe fine est activée. Sinon, il est vérifié si une valeur valide a été adoptée pour le commutateur booléen.

Si aucune des deux commandes n'a été adoptée, elles seront réglées et activées à l'aide de `\FamilyKeyStateProcessed`. Si le test de charge échoue également, l'échec de `\FamilySetBool` est signalé et enregistré à la fin de l'exécution. La déclaration utilisée dans les tests `\ifstr` est expliquée [section 11.3](#).

Exemple : un pré-réglage par défaut désigne le type de coupe à utiliser qui doit être grossière ou fine. même s'il n'est pas précisé comment les coupes doivent couper :

```

\@Ifundefined{if@coupe fine}{%
\expandafter
\newif \csname if@coupe fine\endcsname}{}%
\@Ifundefined {if@coupe}{%
\expandafter
\newif \csname {if@tranché}\endcsname}{%
\DefineFamilyKey{boucher}%
{coupe}[true]{%
\FamilySetBool{boucher}{coupe}%
{coupe}%
{#1}%
\Ifx \FamilyKeyState\FamilyKeyStateProcessed
\@{coupe fine}{false}
\else
\ifstr{#1}{fin}{%
\@{jambon}{true}
\@{coupe fine}{true}

```

```
\FamilyKeyStateProcessed  
}{}%  
\fi  
}%
```

```
\FamilyElseValues
```

Dans les versions antérieures de *scrbase* vous pouviez utiliser d'autres valeurs avec la commande `\FamilyElseValues` pour un traitement et les stocker sous la forme :

```
'valeur', 'valeur'...
```

par `\FamilyUnknowValue` qui pouvait diffuser alors un message d'erreur. Depuis la version v3.12 `\FamilyUnknowValue` ne signale plus les messages d'erreur et utilise `\FamilyKeyState`. En conséquences, l'utilisation de `\familyElseValue` est dépassée mais elle est détectée et elle conduit à un appel d'ajustement du code en conséquence.

## 11.4 Exécution conditionnelle de *scrbase*

Le paq *scrbase* fournit plusieurs commandes d'exécution optatives différentes de la syntaxe de  $\TeX$ , construites telles que :

```
\iftrue ...  
\else ...  
\fi
```

dont la syntaxe est utilisée avec des arguments connus par les commandes  $\LaTeX$  comme `\IfFileExists`, `\@ifundefined`, `\@ifpackloaded` et bien d'autres.

Cependant, quelques auteurs de paqs préfèrent la syntaxe de  $\TeX$ , même pour les utilisateurs de l'interface  $\LaTeX$ , ce qui peut inévitablement conduire à des conflits de noms avec *scrbase*, la probabilité de l'utilisation d'instructions quasi identiques étant très forte, même avec une sémantique analogue, et poser des problèmes en raison de la syntaxe différente. En fait, les conditionnelles de *scrbase* sont basiques : certaines instructions sont définies par *scrbase*, d'abord en tant que commandes internes avec `\scr@`, les commandes sans ce préfixe peuvent être définies ultérieurement, ce qui peut être évité en utilisant l'option `internalOnly` (voir la section 11.1).

Les auteurs de paqs et de classes devraient utiliser les commandes internes de *KOMA-Script*, et les instructions d'utilisation qui les accompagnent.

```
\scr@ifundefinedorrelax{nom}{instructions}{autres instructions}  
\ifundefinedorrelax{nom}{puis instructions}{autres instructions}
```

Cette commande fonctionne essentiellement comme `\@ifundefined` du noyau L<sup>A</sup>T<sub>E</sub>X (voir [Bra+05]). Les instructions sont exécutées lorsque le nom de la commande est indéfini ou actuellement `\`. Contrairement à `\@ifundefined`, ce nom, s'il était précédemment indéfini, n'est pas défini `\relax` par défaut. En outre, l'utilisation de  $\epsilon$ -T<sub>E</sub>X n'a pas d'effet mémoire.

```
\scr@ifpdftex{alors des instructions}{et d'autre part}  
\ifpdftex{puis la partie}{et d'autre part}
```

Si pdfT<sub>E</sub>X a été utilisé, les instructions sont exécutées sinon l'instruction `else` par défaut. Il n'est pas important de savoir si le fichier PDF est généré, donnée rarement vraiment utile. Mais en général, vous devriez tester davantage la commande souhaitée.

```
\scr@ifVtex{alors l'instruction}{ou l'instruction else}  
\ifVtex{alors l'instruction}{ou l'instruction else}
```

Si VT<sub>E</sub>X a été utilisé, les instructions sont exécutées sinon l'instruction `else` par défaut. Ce test est rarement utile. Par contre, vous devriez tester davantage l'instruction souhaitée. (voir précédente observation)

```
\scr@ifpdfoutput{alors l'instruction}{ou l'instruction else}  
\ifpdfoutput{alors l'instruction}{ou l'instruction else}
```

Si un fichier PDF est créé, alors les instructions sont exécutées, sinon la partie `else`. Il importe peu que le fichier PDF soit généré à l'aide de pdfT<sub>E</sub>X, de VT<sub>E</sub>X ou d'un autre moteur de T<sub>E</sub>X.

```
\scr@ifpsoutput{alors des instructions}{et d'autre part}  
\ifpsoutput{puis la partie}{et d'autres instructions}
```

Si un fichier PostScript est créé, les instructions sont exécutées, sinon les instructions `else`. VT<sub>E</sub>X peut directement générer un PostScript qui sera, ici, reconnu. Si VT<sub>E</sub>X n'est pas utilisé, on peut définir un bouton avec `\if@dvi`, et la décision dépendra de ce commutateur fourni par KOMA-Script dans le paquet `typearea`.

```
\scr@ifdvioutput{des instructions}{et d'autre part}  
\ifdvioutput{puis la partie}{et puis aussi}
```

Lors de la génération d'un fichier DVI, les instructions sont exécutées, sinon l'instruction `else`. Un fichier DVI est généré quand aucune sortie directe vers un fichier PDF ou PostScript ne peut être reconnue.



```
\nom{alors l'instruction}{puis la partie}
```

$\varepsilon$ -TeX sera utilisé pour tester si une commande avec un nom donné a déjà été définie ou pas. Si la commande est définie, les instructions seront ensuite exécutées. Il n'existe pas d'option interne à cette déclaration.

```
ifstr{chaîne}{chaîne}{alors l'instruction}{sinon l'instruction  
else}
```

Les deux arguments de chaîne sont développés et ensuite comparés.

Avec les mêmes extensions, les instructions sont réalisées, sinon l'instruction `else` est exécutée. La commande n'est pas complètement extensible et il n'existe pas d'option interne à cette déclaration.

```
ifstrstart{chaîne}{chaîne}{alors l'instruction}{ou l'instruction  
else}
```

Les deux arguments de chaîne sont développés et ensuite comparés. Si la première chaîne débute en dehors de l'espace avec la deuxième chaîne, alors l'instruction est appliquée sinon l'instruction `else` sera exécutée. Les expressions invalides provoquent un message d'erreur. La commande n'est pas extensible complètement et il n'existe pas d'instruction interne correspondante.

```
ifisdimen{code}{alors l'instruction}{ou l'instruction else}
```

Si le résultat de l'expansion du code dans `\dimen`, est reconnu comme valeur de longueur de TeX, alors les instructions seront réalisées, sinon l'instruction `else` sera exécutée. Les expressions invalides provoquent un message d'erreur. La commande n'est pas extensible et il n'existe pas d'instruction interne.

```
ifisdimension{code}{alors l'instruction}{l'instruction else}
```

Si le résultat de l'expansion du code dans un `\dimexpr` est reconnu expression d'une longueur TeX, alors les instructions seront réalisées, sinon l'instruction `else` sera exécutée. Les expressions invalides provoquent un message d'erreur. La commande n'est pas complètement extensible et il n'existe pas d'instruction interne correspondante.

```
ifisdimexpr{code}{alors l'instruction}{ou l'instruction else}
```

Si le résultat de l'expansion du code dans `\skip`, est reconnu comme registre à distance de TeX, alors les instructions seront réalisées, sinon l'instruction `else` est exécutée. Les expressions invalides provoquent un message d'erreur. La commande n'est pas extensible et il n'existe pas d'instruction interne.

```
ifisskip{code}{alors l'instruction}{ou l'instruction else}
```

Quand l'expression du code `\glue` correspond à la syntaxe de la valeur d'une distance, la première partie est exécutée, sinon la partie de l'instruction `else`. Une commande invalide provoque un message d'erreur. La déclaration n'est pas entièrement extensible et il n'en existe pas de variation interne.

```
ifisglue{code}{alors l'instruction}{ou l'instruction else}
```

Si le code de l'expansion donne un résultat dans `\glueexpr` reconnu comme expression de distance par  $\TeX$ , les instructions sont réalisées, sinon `else` est exécutée. Les expressions invalides provoquent un message d'erreur.

La commande n'est pas complètement extensible. Il n'existe pas d'instruction interne correspondante.

```
ifisglueexpr{code}{alors l'instruction}{ou l'instruction else}
```

Si le code de l'expansion donne un résultat identifié « registre de compteur » de  $\TeX$  dans `\count`, les instructions sont menées à bien, sinon l'instruction `else` sera appliquée. La commande n'est pas complètement extensible et il n'existe pas de commande interne correspondante. Pour le test avec des compteurs de  $\LaTeX$ , voir `\ifiscounter`.

```
ifiscount{code}{alors l'instruction}{ou l'instruction else}
```

Lorsque le résultat de l'expansion du code correspondant à la syntaxe de la valeur d'un compteur est un nombre entier positif ou négatif, la commande est effectuée, sinon l'instruction `else` sera appliquée. La déclaration n'est pas entièrement extensible et il n'existe pas de commande interne correspondante.

```
ifisnumber{code}{alors l'instruction}{ou l'instruction else}
```

Si le code dans un `\numexpr` est le résultat d'une expression de nombre  $\TeX$ , alors les instructions sont exécutées, sinon l'instruction `else` est utilisée. Les expressions défectueuses mènent à un message d'erreur. La version n'est pas entièrement extensible et il n'existe pas de commande interne correspondante.

```
ifisnumexpr{code}{alors l'instruction}{ou l'instruction else}
```

Si le compteur est identifié comme un compteur  $\LaTeX$  les instructions sont exécutées, sinon l'instruction `else` est utilisée. La version n'est pas entièrement extensible et il n'existe pas de commande interne analogue.

```
ifiscounter{compteur}{alors l'instruction}{ou l'instruction else}
```

Ici, aucun nombre n'est comparé et les instructions sont exécutées lorsque la chaîne ne comporte que des chiffres. À défaut, l'instruction `else` est appliquée. Il n'existe pas de commande interne correspondante à la déclaration.

```
ifnumber{chaîne}{alors l'instruction}{ou l'instruction else}
```

Ici aucun décompte n'est comparé. La partie fonctionne quand l'expansion de la chaîne se compose uniquement de chiffres, sinon, la partie `else` sera utilisée. Il n'existe pas de variation interne de cette déclaration,

```
ifdimen{chaîne}{alors l'instruction}{ou l'instruction else}
```

Ici, aucune longueur n'est comparée, la pièce s'exécute quand l'extension de la chaîne est une longueur valide avec une unité de longueur valable, sinon la partie `else` est utilisée. Il n'existe pas de variation interne de cette déclaration,

```
if@document instructions\alors l'instruction \fi
```

Cette branche existe comme déclaration interne, `\if@document` correspond au préambule du document. Ensuite, après `\begin{document}` se placent les conditionnelles `\iffalse` et `\iftrue`. Les classes et les auteurs peuvent utiliser ces instructions indispensables quand existe une différence de comportement entre préambule et corps du document. Il est important de noter que cette branche figure dans la syntaxe de  $\text{T}_{\text{E}}\text{X}$  et non dans celle de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  pour cette déclaration.

## 11.5 Identificateur de la langue

Normalement, il faut modifier ou définir les termes dépendant de la langue tels que `\captions` en français de manière à ce que les nouveaux termes soient définis ou redéfinis, en plus des conditions disponibles, ce qui paraît d'autant plus difficile que certains paqs comme `german` ou `ngerman` redéfinissent ces paramètres quand ils sont chargés, en supprimant les modifications apportées précédemment.

Il paraît donc logique de retarder les changements avec `\AtBeginDocument` jusqu'à `\begin{document}`, le chargement du paq est ensuite terminé. On peut redéfinir les termes dépendant de la langue après `\begin{document}`, ou ne pas les mettre du tout dans le préambule.

Un autre inconvénient est que certains paqs supplémentaires dépendant aussi de la langue définissent les termes dans `\captionsSprache` (la langue de la légende, par exemple), alors que d'autres utilisent `\extrasSprache` (langue supplémentaire). Ainsi, l'utilisateur doit être très exactement informé afin de compléter de la bonne manière une instruction correcte.

Le paq `scrbase` fournit quelques commandes supplémentaires pour ce faire.

```
\defcaptionname{liste des langages}{terme}{definition}
\providcaptionname{liste des langages}{terme}{definition}
\newcaptionname{liste des langages}{terme}{definition}
\renewcaptionname{liste des langages}{terme}{definition}
\defcaptionname*{liste des langages}{terme}{definition}
\providcaptionname*{liste des langages}{terme}{definition}
\newcaptionname*{liste des langages}{terme}{definition}
\renewcaptionname*{liste des langages}{terme}{definition}
```

L'utilisation d'une commande de synthèse permet d'assigner une définition en fonction de la langue particulière à un terme. Plusieurs langues séparées par une virgule peuvent être spécifiées dans une liste de langages.

Le terme est toujours une macro. Les commandes varient selon qu'une langue donnée ou qu'un terme dans une langue donnée sont déjà définis ou non au moment où la commande est appelée.

Si une langue n'est pas définie, `\providcaptionname` ne fait rien d'autre que d'écrire un message dans le fichier journal. Cela ne se produit qu'une seule fois pour chaque langue. Si une langue est définie, mais sans concept correspondant, elle sera définie en utilisant le contenu spécifié. Le terme ne sera pas redéfini si la langue a déjà examiné une définition ; et un message approprié sera écrit, à la place, dans le fichier journal.

La commande `\newcaptionname` a un comportement légèrement différent. Si une langue n'est pas encore définie, une instruction correspondante sera créée et un message écrit dans le fichier journal. Pour la langue `USenglish`, par exemple, ce serait la commande `\captionsUSenglish` ou `\captionsngerman` pour la langue `ngerman`. Si un terme n'est pas encore disponible dans cette langue, il est défini par le contenu souhaité. S'il existe déjà dans une langue, cela se traduira par un message d'erreur.

La commande `\renewcaptionname` se comporte à nouveau différemment. Si la langue n'est pas définie, un message d'erreur est émis. Toutefois, si la langue est définie et que le terme est inconnu dans cette langue, un message d'erreur est également émis. Sinon, le terme de la langue sera redéfini selon la définition avec le contenu souhaité.

KOMA-Script emploie `\providcaptionname` pour définir les commandes dans la section 21.4.

Si vous préférez « fig. » au lieu de « figure » avec `ou` ou `french` de `babel`, vous pouvez l'obtenir à l'aide de :

```
\renewcaptionname{french}{figurename}{fig}
```

Si vous désirez le même changement non seulement dans *ngerman* mais aussi dans *nswissgerman*, ou encore dans *Österreichisch* et *Schweizer Deutsch*, vous n'avez pas besoin d'un supplément :

```
\renewcaptionname{naustrian}{figurename}{Abb.}
\renewcaptionname{nswissgerman}{figurename}{Abb.}
\renewcaptionname{UKenglish}{figurename}{figure.}
```

et la liste des langues peut être facilement étendue :

```
\renewcaptionname{ngerman,nswissgerman}{figurename}{Abb.}
\renewcaptionname{USenglish,UKenglish}{figurename}{fig.}
```

Depuis KOMA-Script 3.12, il n'est plus nécessaire de retarder la définition ou la redéfinition utilisant `\AtBeginDocument` à `\begin{document}`. Cela est fait par `scrbase` même appelé dans le préambule.

En outre `scrbase` vérifie également si une nouvelle définition des termes existe dans `\captionlanguage` ou dans `\extralanguage`. Les versions étoilées des commandes utilisent avant tout les langues supplémentaires dont les définitions se trouvent généralement après l'application `\captionlanguage`.

Bien que la redéfinition des identifiants propres à la langue du paq fonctionne généralement bien avec `hyperref`, penser à utiliser la langue supplémentaire `\extralanguage`.

Les classes sont habituellement définies en langue anglaise et les paqs sont affichés et décrits dans le tableau 12.1.

## 11.6 Identification de KOMA-Script

Si le paq `scrbase` peut être utilisé indépendamment de KOMA-Script avec d'autres paqs et classes, il est néanmoins un paq KOMA-Script, et pour cette raison fournit des commandes pour identifier KOMA-Script et s'identifier comme un paq KOMA-Script

<code>\KOMAScript</code>
--------------------------

Cette commande définit uniquement le mot `KOMA-Script` avec utilisation de la police `sans-serif` et une fixation des parties en capitales dans l'ensemble. Tous les paqs et les classes de `KOMA-classes` spécifient cette commande, si elle n'a pas déjà été précisée.

TABLE 11.1 – Ensemble de termes habituels dépendant du paq de langue

<code>\abstractname</code>	rubrique pour le résumé des noms abstraits
<code>\alsoname</code>	autre appellation pour « voir aussi » dans les références croisées supplémentaires
<code>\appendixname</code>	titre « annexe » de la tête de chapitre d'une pièce jointe
<code>\bibname</code>	titre de la bibliographie
<code>\ccname</code>	préfixe de la rubrique liste de distribution d'une lettre
<code>\chaptername</code>	nom du « chapitre » dans le titre d'un chapitre
<code>\contentsname</code>	nom de rubrique de la table des matières
<code>\enclname</code>	préfixes de rubriques incluses dans une lettre
<code>\figurename</code>	nom de la rubrique signature
<code>\glossaryname</code>	nom de la rubrique glossaire
<code>\headtoname</code>	nom de l'entête des pages de lettres
<code>\indexname</code>	index des mots-clés
<code>\listfigurename</code>	liste des figures
<code>\listtablename</code>	liste des tables
<code>\pagename</code>	page dans la pagination de lettres
<code>\partname</code>	« partie » dans le titre d'une partie
<code>\prefacename</code>	rubrique de la préface
<code>\proofname</code>	nom de l'argument préfixe
<code>\refname</code>	liste des références
<code>\seename</code>	« voir » les références croisées de l'index
<code>\tablename</code>	nom des légendes en préfixe de la rubrique table

La définition est donnée par `\DeclareRobustCommand`.

`\KOMAScriptVersion`

Avec cette commande, KOMA-Script définit sa version principale qui prend toujours la forme de la version KOMA-Script actualisée. Cette version principale est la même pour tous les paqs et toutes les classes de KOMA-Script. Pour cette raison, elle peut aussi être appelée après le chargement de `scrbase`. Par

exemple, ce document a été écrit en utilisant la version KOMA-Script v3.21 du 14 juin 2016.

## 11.7 Extensions du noyau L<sup>A</sup>T<sub>E</sub>X

Dans certains cas, le noyau de L<sup>A</sup>T<sub>E</sub>X fournit lui-même des instructions qui sont parfois incomplètes et des commandes similaires seraient souvent utiles. Certaines sont fournies par `scrbase` pour les auteurs de paqs et de classes.

```
\ClassInfoNoLine{nom de la classe}{informations}
\PackageInfoNoLine{nom du paq}{informations}
```

Le noyau de L<sup>A</sup>T<sub>E</sub>X fournit à l'auteur de paqs et de classes des commandes telles que `PackageInfo` et `ClassInfo` pour noter des informations comportant le numéro de la ligne courante dans le fichier journal. Il propose `\PackageWarning` et `\ClassWarning`, qui émettent des avertissements avec le numéro de la ligne actuelle. Les commandes `\PackageWarningNoLine` et `\ClassWarningNoLine` émettent des avertissements sans numéro de ligne. Les commandes pour noter des informations sans numéro de ligne dans le fichier journal manquent mais `\ClassInfoNoLine` et `\PackageInfoNoLine` sont fournies par `scrbase`.

```
\l@addto@macro{commande}{extension}
```

Ce noyau fournit une commande interne permettant d'étendre la définition d'une macro-instruction par l'extension globale du code, `\g@addto@macro`. Cela fonctionne, dans ce formulaire, uniquement pour les macros sans arguments. Pourtant, on pourrait avoir besoin de cette déclaration dans certains cas, sous une forme qui fonctionne globalement. Elle est alors fournie par `scrbase` sous la forme de `\l@addto@macro`. Une alternative peut être l'utilisation du paq `etoolbox` qui offre un certain nombre de ces instructions à des fins différentes (voir [Leh11]).

## 11.8 Compétences mathématiques de $\epsilon$ -T<sub>E</sub>X

$\epsilon$ -T<sub>E</sub>X qui est utilisé par L<sup>A</sup>T<sub>E</sub>X et nécessairement par KOMA-Script, placé avant `\numexpr`, a sa fonctionnalité étendue pour le calcul de l'arithmétique simple avec des compteurs T<sub>E</sub>X et entiers.

Les quatre opérations et les supports arithmétiques de base sont pris en charge et les arrondis sont corrects en division, mais des opérateurs supplémentaires seraient parfois utiles.

<code>\XdivY{dividende}{diviseur}</code>
<code>\XmodY{dividende}{diviseur}</code>

La commande `\XdivY` renvoie la valeur du quotient entier, la commande `\XmodY` donne la valeur du reste. Cette division est définie par l'équation :

$$\text{dividende} = \text{diviseur} \cdot \text{quotient entier} + \text{reste}^1$$

selon laquelle le dividende et le reste sont des entiers, le reste est supérieur ou égal au diviseur, et le diviseur est un nombre naturel supérieur à 0. La valeur peut être utilisée pour l'affectation à un compteur ou directement dans `\numexpr`. La valeur de sortie à partir de chiffres arabes doit être préfixée par `\the`.

---

1. PAPIRAY précise : je voulais commenter l'équation et finalement, je renonce !



# Chapitre 12

## Dépendances des paqs avec `scrfile`

L'introduction de  $\text{\LaTeX} 2_{\epsilon}$  en 1994 a apporté de nombreuses innovations dans l'utilisation d'extensions de  $\text{\LaTeX}$ . L'auteur du paq dispose aujourd'hui d'une série de commandes pour déterminer si un autre paq, une classe ou certaines options sont utilisées.

Le paq auteur peut charger d'autres paqs ou peut spécifier si des options en cours de route prévoient d'être chargées plus tard. Cela a conduit, entre autres, à tester le chargement des paqs dans n'importe quel ordre en pensant que ce serait sans importance, supposition qui, malheureusement, s'est avérée inexacte.

### 12.1 À propos des dépendances des paqs

De plus en plus souvent, une même commande est définie ou redéfinie par des paqs. L'ordre dans lequel ils sont chargés devient très important, ce qui dans certains cas rend difficile la compréhension de leur comportement et n'incite pas l'utilisateur à réagir. Il est même souvent déconseillé d'intervenir lorsqu'un paq est chargé, ce qui n'est pas toujours facile.

Prenons un exemple simple, l'emploi du paq `longtable` lors de l'utilisation de `KOMA-Script`.

Le paq `longtable` définit ses propres rubriques qui s'adaptent parfaitement aux classes `standard` mais les légendes créées à l'aide de documents `KOMA-Script` sont totalement inadaptées et ne répondent pas aux options de configuration appropriées. Afin de résoudre ce problème, les commandes du paq `longtable`, responsables des légendes de table, doivent être redéfinies.

Cependant, au moment où le paq `longtable` est chargé, la classe `KOMA-Script` a déjà été traitée. Jusqu'à présent, la seule façon de résoudre ce problème pour `KOMA-Script` était de retarder la redéfinition jusqu'au début du document avec l'aide de la macro `\AtBeginDocument`.

Si l'utilisateur souhaite modifier les définitions correspondantes, il est possible de le faire dans le préambule du document, mais cette opération est irréaliste, car plus tard, lors de `\begin{document}` `KOMA-Script` écrasera ces définitions qu'il remplacera par les siennes. Il suffirait de retarder l'exécution jusqu'à ce que le paq `longtable` ait été chargé. Malheureusement, le noyau de `LATEX` ne définit pas de commandes appropriées.

Le paq `scrfile` apporte ici une solution.

De même, il pourrait être concevable que, avant de charger un paq, vous souhaitiez enregistrer la définition d'une macro dans un aide-macro, afin de restaurer son sens après que le paq ait été chargé.

Le paq `scrfile` le permet aussi.

L'utilisation de `scrfile` n'est pas limitée à la dépendance des paqs. Celles à d'autres fichiers peuvent être prises en considération. Par exemple, l'utilisateur peut être averti si le fichier non critique `french.ldf` a bien été chargé.

Bien que ce paq intéresse, en premier lieu, les auteurs de paqs, il existe des cours d'applications pour les utilisateurs ordinaires de `LATEX`. Les exemples donnés dans ce chapitre sont à destination des deux groupes.

## 12.2 Actions avant et après le chargement

`scrfile` peut exécuter des actions avant et après le chargement des fichiers. Dans les commandes utilisées pour ce faire, des distinctions sont faites entre les fichiers généraux, les classes et paqs.

```
\BeforeFile{fichier}{instructions}
\AfterFile{fichier}{instructions}
```

La macro `\BeforeFile` assure que les instructions sont exécutées avant le prochain chargement d'un fichier de macro particulier.

`\AfterFile` fonctionne d'une manière similaire, et les instructions ne sont exécutées qu'après le chargement du fichier. Si le fichier n'est pas chargé, les instructions dans les deux cas, bien sûr, ne seront jamais exécutées.

Pour mettre en œuvre ces fonctionnalités, `scrfile` redéfinit la commande  $\LaTeX$  bien connue `\InputIfFileExists`. Au cas où cette définition n'est pas prévue, `scrfile` émet un avertissement. Ceci est fait quand la déclaration dans les versions ultérieures de  $\LaTeX$  est modifiée ou redéfinie par un autre paq. La commande `\InputIfFileExists` est utilisée par  $\LaTeX$  pour charger un fichier. Cela se produit indépendamment du fait que les fichiers `\LoadClass`, `\documentclass`, `\usepackage \RequirePackage`, `\include` soient chargés, ou des instructions similaires.

Seul `\input{foo}` charge le fichier sans utiliser `\InputIfFileExists`.

Vous devriez toujours utiliser `\input{foo}`. Noter les accolades autour du nom de fichier.

```
\BeforeClass{classe}{instructions}
\BeforePackage{paq}{instructions}
```

Ces deux commandes fonctionnent d'une manière similaire à `\BeforeFile`. La seule différence est que la classe ou le paq sont spécifiés avec leur nom et pas avec le nom de fichier. Les terminaisons `cls` et `sty` peuvent également être omises.

```
\AfterClass{classe}{instructions}
\AfterClass*{instructions}
\AfterClass+{instructions}
\AfterClass"! {classe}{instructions}
\AfterAtEndOfClass{classe}{instructions}
\AfterPackage{paq}{instructions}
\AfterPackage*{paq}{instructions}
\AfterPackage+{paq}{instructions}
\AfterPackage"! {paq}{instructions}
\AfterAtEndOfPackage{paq}{instructions}
```

Les commandes `\AfterClass` et `\AfterPackage` opèrent en grande partie de la même manière que `\AfterFile` avec l'infime différence que la classe ou le paq

est répertorié avec son nom et pas avec le nom de fichier. Les terminaisons `.cls` et `.sty` sont aussi omises ici. Une fonctionnalité supplémentaire existe dans les versions étoilées. Elle exécute les instructions non seulement à la prochaine fois que la classe correspondante ou le paq approprié est chargé, mais aussi immédiatement si le paq a été déjà chargé.

Dans la version `+`, les instructions sont exécutées lorsque la classe ou le paq a été entièrement chargé. La différence avec la version étoilée n'est valable que si le chargement de la classe ou du paq a commencé, mais n'est pas terminé. Elles seront exécutées avant les instructions `\AtEndOfClass` ou `\AtEndOfPackage` alors que le chargement de la classe ou du paq n'est pas encore terminé.

Si `\AtEndOfClass` est utilisé par une classe ou `\AtEndOfPackage` par un paq pour exécuter des instructions après que les fichiers de la classe ou du paq ont été chargés, et si vous voulez exécuter de nouvelles instructions après celles de ces commandes, vous pouvez utiliser les versions avec un point d'exclamation, `\AfterClass"!` et `\AfterPackage"!`, mais, dans ce cas, les instructions ne sont pas exécutées dans le contexte de la classe ou du paq spécifié. Si vous voulez, juste au cas où la classe ou le paq n'a pas été chargé, vous assurer que les instructions sont bien exécutées en dehors du contexte de la classe ou du paq spécifié, utiliser pour les classes la commande `\AfterAtEndOfClass` et pour les paqs `\AfterAtEndOfPackage`.

Dans ce qui suit, un exemple de classes et de paq est donné pour les auteurs. Il montre comment KOMA-Script lui-même emploie les nouvelles commandes. La classe `scrbook` contient :

```
\AfterPackage{hyperref}{%
\@ifpackagelater{hyperref}{2001/02/19}{}{%
\ClassWarningNoLine{scrbook}{%
Vous utilisez une ancienne version hyperref qui provoque le
dysfonctionnement de nombreux pilotes et produit un comportement
étrange de \string\addchap\space. Mettre à jour hyperref avec la
dernière version 6.83q - au 2016/07/25}%
}%
}
```

Les anciennes versions du paq `hyperref` définissent une macro de la classe `scrbook` de telle manière que son exécution ne fonctionne pas avec les nouvelles versions de KOMA-Script.

Les nouvelles versions s'abstiennent de faire des changements si une version plus récente de KOMA-Script est identifiée. Dans l'hypothèse où `hyperref` est

chargé ultérieurement, `scrbook` effectue immédiatement après le chargement un contrôle du paq pour vérifier si la version est compatible.

Si ce n'est pas le cas, un avertissement est émis.

Les trois classes KOMA-Script utilisent la macro suivante :

```
\AfterPackage{caption2}{%
\renewcommand*{\setcapindent}{%
```

Après le chargement du paq `caption2`, KOMA-Script redéfinit sa commande `\setcapindent`. Le code exact de la redéfinition n'est pas important pour cet exemple. On peut noter que `caption2` prend le contrôle de la commande `\caption-légende` et que, par conséquent, la définition normale de la macro `\setcapindent` n'a plus aucun effet. La redéfinition améliore la collaboration avec `caption2`.

Il existe des exemples de l'utilisation pertinente des nouvelles instructions pour l'utilisateur de L<sup>A</sup>T<sub>E</sub>X normal.

Supposons que vous créez un document qui devrait être disponible sous forme de fichier PS, en utilisant L<sup>A</sup>T<sub>E</sub>X et `dvips`, ainsi que sous forme d'un fichier PDF, en utilisant `pdfLATEX`. Le document devrait également contenir des liens hypertextes. Dans la table des matières, vous avez des entrées qui s'étendent sur plusieurs lignes, ce qui n'est pas un problème pour la méthode de `pdfLATEX` car ici des hyperliens peuvent aussi être répartis sur plusieurs lignes, ce qui est impossible si vous utilisez un pilote de `hyperref` tel que `dvips` ou `hypertex`.

Dans ce cas, on adopte `linktocpage` pour la configuration `hyperref` et la décision d'utiliser le pilote `hyperref` se fait automatiquement en s'appuyant sur `hyperref.cfg`. Le fichier aura, par exemple, le contenu suivant :

```
\ProvidesFile{hyperref.cfg}
\@ifundefined{pdfoutput}{\ExecuteOptions{dvips}}
{\ExecuteOptions{pdfTeX}}
\endinput
```

Tout le reste peut maintenant être ignoré et laissé à `\AfterFile` :

```
\documentclass{article}
\usepackage{scrfile,french}
\AfterFile{hdvips.def}{\hypersetup{linktocpage}}
\AfterFile{hypertex.def}{\hypersetup{linktocpage}}
\usepackage{hyperref}
\begin{document}
\listoffigures
```

```
\clearpage
\begin{figure}
\caption{Ceci est un exemple pour une légende assez longue
qui n'utilise pas l'argument optionnel qui permettrait d'en
écrire une plus courte dans la liste des numéros.}
\end{figure}
\end{document}
```

Si maintenant les pilotes de `hyperref`, `hypertex` ou `dvips` sont utilisés, c'est la fonction `linktocpage` de `hyperref` sera appliquée. Dans le cas de `\pdf1`, l'option ne sera pas réglée, puisqu'un autre pilote `hyperref`, `hpdftex.def`, sera utilisé. Cela signifie que ni le fichier `hdvips.def`, ni `hypertex.def` ne seront chargés.

Par ailleurs, vous pouvez insérer le chargement `scrfile` et les commandes `\AfterFile` ci-dessus peuvent être écrites dans un `hyperref.cfg` privé. Si vous le faites, il vaut mieux utiliser la macro `\RequirePackage` plutôt que `\usepackage` (voir [Tea06]).

Les nouvelles lignes doivent être insérée après la ligne `\ProvidesFile`, et immédiatement avant l'exécution des options `dvips` ou `pdftex`.

```
\{instructions}
\AfterReadingMainAux{instructions}
```

Ces commandes diffèrent dans le détail des instructions précédentes car elles permettent des actions avant et après le chargement des fichiers, pour les auteurs de paqs, notamment, qui ont souvent un problème d'écriture dans le fichier auxiliaire `file.aux` après que la dernière page du document a été éditée. Dans l'ignorance de ce problème, le code suivant est fréquemment utilisé comme suit :

```
\AtEndDocument{%
  \if@files
  \write\@auxout{%
    \protect\writethistoaux%
  }%
  \fi
}
```

Cette solution n'est pas idéale pour répondre au problème. Si la dernière page du document a déjà été expédiée avant `\end{document}`, ce code n'entraînera pas d'écriture dans le fichier `.aux`. Si vous essayez de résoudre ce nouveau problème en utilisant `\immediate` juste avant `\write`, le problème inverse se produit : si

une dernière page n'a pas été éditée avant `\end{document}`, `\writethistoaux` est écrit trop tôt dans le fichier `.aux`.

Une autre suggestion proposée comme solution pour ce problème est :

```
\AtEndDocument{%
  \if@filesw
  \clearpage
  \immediate\write\@auxout{%
    \protect\writethistoaux%
  }%
  \fi
}
```

Cette suggestion présente un nouvel inconvénient : la sortie de la dernière page a été forcée par `\clearpage`. Après cela, des instructions comme :

```
\AtEndDocument{%
  \par\vspace*{\fill}%
  Note a la fin du document. \par
}
```

qui ne provoquent plus un avis émis à la fin de la dernière page du document, mais plutôt à la fin de la page suivante. Dans le même esprit, `\writethistoaux` serait écrit une page trop tôt dans le fichier auxiliaire.

La meilleure solution au problème serait de pouvoir tout simplement écrire dans le fichier auxiliaire `.aux` immédiatement après le `\clearpage` final dans `\end{document}`, mais juste avant la fermeture du fichier `.aux`.

C'est l'objectif de `\` :

```
\BeforeClosingMainAux{%
  \if@filesw
  \immediate\write\@auxout{%
    \protect\writethistoaux%
  }%
  \fi
}
```

C'est également réussi lorsque le résultat `\end{document}` de `\clearpage` ne génère pas une sortie d'écran avec une utilisation correcte ou dans l'ignorance des problèmes évoqués ci-dessus, et si `\clearpage` est inséré en argument de `\AtEndDocument`.

Mais il existe une restriction dans l'utilisation de `\` : Vous ne devez pas utiliser d'instruction composée à l'intérieur des instructions de `\`. Si la restriction est ignorée, le résultat en serait aussi imprévisible que les résultats des suggestions problématiques qui utilisent l'ascendance `\AtEndDocument`.

La commande `\AfterReadingMainAux` exécute même les instructions après la fermeture et la lecture du fichier au sein `\end{document}`. Ceci n'est utile que dans quelques cas très rares, par exemple, afficher pour le fichier journal des informations statistiques qui ne sont valides qu'après avoir lu le fichier `.aux`, ou pour exécuter les demandes de rediffusions supplémentaires.

Les instructions `typeset` sont encore plus critiques dans ces instructions qu'à l'intérieur de l'argument de `\BeforeClosingMainAux`.

## 12.3 Remplacer les fichiers lors de la lecture

Dans les sections précédentes de ce chapitre, les commandes sont décrites, avec lesquelles il est possible d'exécuter des instructions avant ou après la lecture d'un fichier particulier, un paq spécifique ou une classe avec `scrfile` qui permet également de lire des fichiers complètement différents.

```
\ReplaceInput{filename}{fichier de remplacement}
```

Cette déclaration définit une substitution du fichier avec le premier argument, nom du fichier source, par le fichier du second argument, nom du fichier de remplacement.

Si `LATEX` doit ensuite charger ce fichier, c'est le fichier de remplacement qui sera appelé. La définition du fichier de remplacement affecte tous les fichiers chargés en utilisant `\InputIfFileExists`, que ce soit par l'utilisateur ou par `LATEX` en interne. Il est nécessaire que `scrfile` redéfinisse `\InputIfFileExists`.

Vous voulez remplacer le fichier `\jobname.aux` par le fichier `\jobname.xua`. Pour ce faire, utiliser :

```
\ReplaceInput{\jobname.aux}{\jobname.xua}
```

Vous voulez maintenant remplacer `\jobname.xua` par `\jobname.uxa` :

```
\ReplaceInput{\jobname.xua}{\jobname.uxa}
```

et finalement remplacer `\jobname.aux` par `\jobname.uxa`, ainsi la chaîne complète de remplacement sera traitée.

Un remplacement dans le circuit :



```
\ReplaceInput{\jobname.aux}{\jobname.xua}
\ReplaceInput{\jobname.xua}{\jobname.aux}
```

se traduirait par une erreur de taille de la sélection. Il n'est donc pas possible de définir le remplacement d'un fichier par lui-même directement ou indirectement. En théorie il serait également possible de remplacer un paq par un autre ou une classe par une autre. Mais, dans ce cas,  $\LaTeX$  reconnaîtrait que le nom de fichier demandé ne correspond pas au nom de l'ensemble ou de la classe et qu'il est erroné. Une solution à ce problème peut être trouvée ci-dessous.

```
\ReplaceClass{classe source}{classe de remplacement}
\ReplacePackage{paq}{paq de remplacement}
```

Une classe ou un paq ne doit jamais utiliser la déclaration `\ReplaceInput` pour remplacer leur nom. Dans ce cas,  $\LaTeX$  signale par un avertissement que le nom de la classe ou le nom du paq ne correspond pas.

Vous décidez inconsidérément de remplacer à la légère le paq `fancyhdr` par `scrpage2` non-conforme :

```
\ReplaceInput{fancyhdr.sty}{scrpage2.sty}
```

Le chargement `fancyhdr` entraînerait un avertissement  $\LaTeX$  :

```
Vous avez demandé \ttt{scrpage2},
                mais le paq fournit \ttt{fancyhdr} !
```

Pour l'utilisateur, cet avertissement serait très déroutant, il n'a pas demandé `scrpage2`, mais `fancyhdr` qui, en fait, a été changé par `scrpage2.sty` en raison de votre définition de substitution.

Une solution à ce problème consiste à remplacer `\ReplaceInput` par l'une des commandes `\ReplaceClass` ou `\ReplacePackage`. Il est important de noter que, comme dans `\documentclass` et `\usepackage` le nom de la classe ou du paq doit être spécifié et non pas le nom de fichier complet.

Le remplacement fonctionne pour les classes chargées avec `\documentclass`, `\LoadClassWithOptions` ou `\LoadClass` et pour les paqs avec `\usepackage`, `\RequirePackageWithOptions` et `\RequirePackage`. Il faut noter que classe et paq de remplacement sont chargés avec les mêmes options que celles requises initialement. Une classe ou un paq remplacé qui ne supporterait pas l'option nécessaire conduirait aux avertissements et messages habituels d'erreurs. Il est toujours possible de redéfinir les options manquantes par des options de remplacement équivalentes en utilisant `\BeforeClass` ou `\BeforePackage`.

Supposons que le paq `oldfoo` doit être remplacé par `newfoo`. Ceci est réalisé par :

```
\ReplacePackage{oldfoo}{newfoo}
```

Supposons que l'ancien paq `oldopt` fournissait une option que le nouveau n'a pas intégrée.

L'utilisation de :

```
\BeforePackage{newfoo}{%
  \DeclareOption{oldopt}{%
    \PackageInfo{newfoo}%
    l'option oldopt n'est pas supportée%
  }}%
```

définit désormais cette option pour le paq `newfoo`. Cela évite de charger le paq `oldfoo` avec une erreur sur l'option `newfoo` qui ne serait pas prise en charge.

Une option `newopt` utilisée à la place de l'option `oldopt` peut aussi être activée par :

```
\BeforePackage{newfoo}{%
  \DeclareOption{oldopt}{%
    \ExecuteOptions{newopt}%
  }}%
```

Un dernier point : il est même possible de spécifier que des nouveaux pré-réglages sont à appliquer à d'autres paqs, lors du chargement :

```
\BeforePackage{newfoo}{%
  \DeclareOption{oldopt}{%
    \ExecuteOptions{newopt}%
  }}%
\PassOptionsToPackage{newdefoptA,newdefoptB}%
{newfoo}%
}
```

ou plus directement :

```
\BeforePackage{newfoo}{%
  \DeclareOption{oldopt}{%
    \ExecuteOptions{newopt}%
  }}%
\PassOptionsToPackage{newdefoptA,newdefoptB}%
```

```
{newfoo}%
```

Pour remplacer une classe, il est nécessaire de charger `scrfile` avant la classe en utilisant `\RequirePackage` au lieu de `\usepackage`.

```
\UnReplaceInput{nom de fichier}
\UnReplacePackage{paq}
\UnReplaceClass{classe}
```

Une définition de remplacement peut également être annulée en utilisant une de ces commandes. La définition de remplacement d'un fichier doit toujours être enlevée en utilisant `\UnReplaceInput`, celle d'un paq avec `\UnReplacePackage` et la définition d'une classe avec `\UnReplaceClass`. L'abolition des instructions de chargement peut entraîner que le fichier, le paq ou la classe d'origine soient chargés au lieu du fichier, du paq ou de la classe de remplacement.

## 12.4 Empêcher le chargement de fichier

Dans les classes et les paqs utilisés au sein des entreprises ou des institutions, il est souvent constaté que de nombreux paqs ne sont pas nécessaires mais néanmoins chargés parce qu'ils sont fréquemment exploités par les utilisateurs. Si un seul de ces paqs chargé automatiquement pose un problème, vous devez en quelque sorte en empêcher le chargement. Encore une fois `scrfile` offre une solution simple.

```
\PreventPackageFromLoading [ici le code]{liste des paqs}
\PreventPackageFromLoading* [ici le code]{liste des paqs}
```

Si cette déclaration est appelée avant de charger un paq avec `\usepackage`, `\RequirePackage` ou `\RequirePackageWithOptions`, le chargement du paq est efficacement empêché s'il se trouve dans la liste des paqs. Vous travaillez, par exemple, dans une société où les documents sont créés avec la police `latin moderne`. La classe de la société FIRMENCI contient les lignes :

```
\RequirePackage[T1]{fontenc}
\RequirePackage{lmodern}
```

Mais maintenant, vous souhaitez utiliser `XqLATEX` ou `LuaLATEX` pour la première fois. Dans ce cas, le chargement de `fontenc` ne serait pas une bonne suggestion et `latin moderne` serait la spécification de police recommandée, police par défaut du paq, alors vous voulez éviter de charger les deux paqs. Cela peut être fait, en chargeant la classe dans votre propre document comme ceci :

```
\RequirePackage{scrfile}
\PreventPackageFromLoading{fontenc,lmodern}
\documentclass{Firmenci}
```

L'exemple ci-dessus montre aussi que le paq `scrfile` peut être chargé avant la classe. Dans ce cas `\RequirePackage` doit être utilisé, plutôt que `\usepackage` qui n'est pas autorisé avant `\documentclass`. Une liste de paqs vide ou un paq déjà chargé lance un avertissement par `\PreventPackageFromLoading`, tandis que `\PreventPackageFromLoading*` écrit seulement un message correspondant dans le fichier journal. L'argument optionnel peut être utilisé pour exécuter du code au lieu de charger le paq. Cependant, aucun autre paq ni fichier ne peuvent être chargés à la place dans le même temps. Pour remplacer un paq par un autre, voir `\ReplacePackage` à la section 12.2. Noter également que le code sera exécuté à chaque fois que vous essayez de charger le paq.

<pre>\StorePreventPackageFromLoading{\commande} \ResetPreventPackageFromLoading</pre>
---

`\commande` définit avec `\StorePreventPackageFromLoading` la liste des paqs dont le chargement doit être évité, mais `\ResetPreventPackageFromLoading` en revanche réinitialise la liste des paqs qui devraient être impossible à charger. Après cela, tous les paqs peuvent être chargés à nouveau.

Pour empêcher qu'un utilisateur bloque le chargement d'un paq dont vous avez vraiment besoin dans votre propre paq, avec `\PreventPackageFromLoading`, il faut anticiper en ajoutant ce paq à la liste des exceptions que vous établissez préalablement au chargement de votre paq :

```
\ResetPreventPackageFromLoading
\RequirePackage{foo}
```

Malheureusement, cela présente l'inconvénient d'effacer la liste complète de toutes les exceptions de l'utilisateur. Pour éviter cela, stocker d'abord cette liste que vous restaurerez à la fin :

```
\newcommand*{\Users@PreventList}{}%
\StorePreventPackageFromLoading\Users@PreventList
\ResetPreventPackageFromLoading
\RequirePackage{foo}
\PreventPackageFromLoading{\Users@PreventList}
```

Il est important de noter que `\Users@PreventList` serait également défini par la commande `\StorePreventPackageFromLoading` même s'il est déjà défini auparavant. En d'autres termes : `\StorePreventPackageFromLoading` écrase

une définition existante. Pour cette raison, `\newcommand*` a été utilisé dans cet exemple pour obtenir un message d'erreur, si `\Users@PreventList` a déjà été défini.

À ce stade, il faut retenir que tout manipulateur de la liste est responsable, s'il utilise `\StorePreventPackageFromLoading`, de sa capacité de restauration. Par exemple, les éléments de la liste doivent être séparés par une virgule, ne doivent pas contenir d'espace blanc ou de groupe entre accolades, et seront entièrement extensibles.

Noter également que `\ResetPreventPackageFromLoading` ne nettoie pas le code d'un paq mais suspend temporairement son exécution qui ne sera pas faite tant que la prévention n'est pas réactivée.

```
\UnPreventpackFromLoading{liste des paqs}
\UnPreventpackFromLoading*{liste des paqs}
```

Au lieu de cela, la liste des paqs dont le chargement doit être empêché est remise à zéro, vous pouvez également supprimer certains paqs de cette liste. La version étoilée de la commande nettoie également le code.

Le code d'une liste restaurée, par exemple, lors de la réactivation de la liste des paqs, ne fonctionne pas toujours.

Pour éviter de charger un paq `foo` car vous ne voulez pas d'un code déjà enregistré et vous préférez exécuter votre propre code, vous le ferez avec :

```
\UnPreventPackageFromLoading*{foo}
\PreventPackageFromLoading[\%
\typeout{code de substitution} %
]{foo}
```

Pour la commande `\UnPreventPackageFromLoading*`, il n'est pas important que les paqs étaient auparavant exclus du chargement. Vous pouvez, à la place, utiliser l'instruction indirectement pour supprimer le code de tous les paqs :

```
\StorePreventPackageFromLoading\TheWholePreventList
\UnPreventPackageFromLoading*{\TheWholePreventList}
\PreventPackageFromLoading{\TheWholePreventList}
```

Dans ce cas, les paqs empêchés d'être chargés le sont toujours mais leur code a été nettoyé et ils ne seront pas exécutés plus longtemps.



# Chapitre 13

## Remplacer et enregistrer les fichiers avec `scrwfile`

Un des problèmes qui n'ont pas été résolus par l'introduction de  $\varepsilon$ -TeX est le fait que seulement 18 fichiers TeX peuvent être ouverts simultanément pour l'écriture, quantité qui, au premier abord, paraît assez grande.

Toutefois, il faut noter que L<sup>A</sup>T<sub>E</sub>X lui-même occupe une partie de ces fichiers : `\@mainaux`, `\@partaux`, `\tableofcontents`, `\listoffigures`, `\listoftables`, `\makeindex` et pour chaque répertoire généré à partir de L<sup>A</sup>T<sub>E</sub>X. À cet effet, les fichiers d'aide proviennent de paqs comme `hyperref` ou `minitoc`.

En fin de compte, il peut donc arriver que, à un moment donné, le message :

```
! No room for a new \write .
\ch@ck..\else \errmessage {No room for a new #3}
\fi
```

apparaisse.

```
(! Pas de place pour une nouvelle \write (écriture).
\ch@ck..\else \errmessage {pas de place pour un nouveau \#3}
\fi)
```

Cette version de L<sup>A</sup>T<sub>E</sub>X ouvre toujours immédiatement un nouveau fichier pour écrire dans les répertoires tels que table des matières, liste des tables et liste des figures, ce qui a également un autre inconvénient.

Ces répertoires sont définis par les commandes correspondantes, mais peuvent également ne pas être fixés une fois de plus, parce que leurs fichiers auxiliaires restent vides jusqu'à la fin du document. Un changement fondamental du noyau de  $\LaTeX$  fourni par le paq `scrwfile` peut résoudre les deux problèmes.

## 13.1 Modification fondamentale du noyau $\LaTeX$

Les classes  $\LaTeX$  utilisent la commande du noyau  $\LaTeX$  `\starttoc` pour ouvrir, par exemple, un répertoire `\tableofcontents` ou `\listoffigures`, liste de flottants ou similaire. Ainsi  $\LaTeX$  lit le fichier d'aide avec le contenu du répertoire, mais l'ouvre aussi en écriture.

Néanmoins, si par la suite, de nouvelles entrées sont effectuées en utilisant, par exemple `\addtocontents` ou `\addcontentsline`, elles ne seront pas écrites immédiatement dans le fichier d'aide.  $\LaTeX$  `\writefile` les enregistre dans un fichier auxiliaire et c'est seulement lors de la relecture des fichiers `.aux`, à la fin du document, que `\writefile` écrit les instructions dans des fichiers auxiliaires réels.

Les fichiers d'aide ne sont pas fermés explicitement par  $\LaTeX$  qui compte sur  $\TeX$  pour les fermer de toute façon à la fin.

Cette procédure garantit que les fichiers auxiliaires seront décrits avec le `\end{document}`, mais sont également ouverts simultanément pendant toute l'utilisation de  $\LaTeX$ , de la première page jusqu'à la fin du document, bien que leur contenu soit écrit après cette fin.

L'approche de `scrwfile` est exactement celle-ci : redéfinition de `\starttoc` et de `\writefile`

Certes, des changements du noyau de  $\LaTeX$  peuvent potentiellement entraîner des incompatibilités avec d'autres paqs. Dans le cas de `scrwfile` cela arrive avec tous les paqs qui redéfinissent (trop) `\starttoc` ou `\writefile`.

Parfois, changer l'ordre de chargement des paqs peut aider.

En fait, de tels problèmes avec `scrwfile` n'ont pas été rapportés récemment, bien que plusieurs utilisateurs aient testé le paq depuis la première version, pendant plus d'un an. Si cela, néanmoins, vous arrivait, merci de contacter l'auteur de `KOMA-Script`.



## 13.2 Le système de fichiers

Le chargement du paq avec `\usepackage{scrwfile}` redéfinit `\@starttoc` de sorte qu'il n'est plus nécessaire d'ouvrir en écriture un fichier ou de lui allouer un descripteur, ces procédures devenant automatiques.

Juste avant la fermeture `\end{document}` du fichier.aux, `\@writefile` redéfinit cette déclaration, non plus dans les fichiers habituels d'aide, mais dans un nouveau fichier auxiliaire de support réel, unique, avec extension `.wrt`. Après avoir lu le fichier `.aux`, ce dossier sera traité une fois par le fichier d'aide.

Après lecture des fichiers `.wrt`, le fichier est toujours en cours de traitement et sera écrit une fois pour chacun des fichiers auxiliaires avec `\@writefile`. Toutefois, il ne doit pas être ouvert en même temps que chacun de ces fichiers auxiliaires. Il ne le sera donc que pour l'écriture et, à chaque fois, de nouveau explicitement fermé.

Depuis qu'un traitement de l'écriture interne à  $\text{\LaTeX}$  est réutilisé, `scrwfile` ne nécessite aucune écriture distincte du fichier pour ce type de répertoire. Même si une seule table des matières doit être générée, après le chargement de `scrwfile` un descripteur de fichier d'écriture sera disponible pour `glossaires`, `index`, `bibliographies`, et autres répertoires qui n'utilisent pas `\@starttoc`.

En outre, le nombre de table des matières et de liste de tout ce qui utilise `\@starttoc` ne sera pas limité.

## 13.3 Le clonage des entrées avec CloneFileWrite

Il est parfois utile d'accéder à un fichier non pas une, mais plusieurs fois. Comme `\@starttoc` ne permet pas d'ouvrir les fichiers pour écrire longuement, cela peut être fait en utilisant tout simplement `\@starttoc` plusieurs fois avec la même extension.

Après l'adaptation de `\@writefile` aux fichiers systèmes précédents afin qu'il n'écrive pas directement le fichier d'aide correspondant, une autre fonction est développée : une copie de l'instruction `\@writefile` dans le fichier `.wrt` peut être transférée à d'autres entrées cibles, ce qui permet de copier l'entrée d'un fichier dans un autre fichier. Ainsi, toutes les entrées du fichier `\jobname.source` peuvent être ajoutées à `\jobname.destination`. Si la destination d'extension est une nouvelle destination, elle sera ajoutée à la liste des extensions connues utilisant le paq `KOMA-Script tocbasic`.

Nous appelons cette opération de copie « clonage ».

```
\TOCclone{titre}{source}{cible}
\listof{cibles}
```

Ces fichiers gèrent les fonctions d'entrées de clonage ainsi que les répertoires entiers à cloner. C'est seulement à la fin du fichier d'aide du répertoire que vous connaissez les éléments qui doivent être copiés. De plus, vous devez spécifier le suffixe d'un fichier cible dans lequel les entrées seront copiées. Bien sûr, vous pouvez ajouter des entrées supplémentaires dans ce répertoire cloné.

L'extension cible du fichier cible est gérée administrativement en utilisant `tocbasic` (voir le chapitre 14). Si un tel fichier est déjà sous contrôle de `tocbasic` un avertissement est émis. Sinon, `tocbasic` crée un nouveau répertoire pour cette extension. Le titre de ce répertoire peut être déterminé avec l'argument en option de l'entête de liste. Ensuite, vous pouvez sortir ce répertoire, par exemple via la commande `\listofdestination` (liste des extensions de cible), et une nouvelle liste de macro est définie. La rubrique sera une `section` (ou un chapitre) de cette liste.

Ce nouveau répertoire peut être produit, par exemple, via la commande `\listofcible`. Le niveau de répertoire des propriétés numérotées `leveldown`, `numbered`, `onecolumn` et `totoc` (voir la commande `\setuptoc section 14.2`) est automatiquement inclus dans le répertoire cible, s'ils étaient déjà définis pour le répertoire source. Dans ce cas, plusieurs caractéristiques `tocbasic` de la source seront copiées à destination, si elles ont été mises en place lorsque `\TOCclone` a été utilisé. Les attributs `nobabel` sont clonés, les commandes de sélection de la langue font partie du fichier d'aide qui l'est systématiquement.

Vous voulez un aperçu de répartition, dans lequel seuls les chapitres sont affichés, en plus de la table des matières habituelle :

```
\usepackage{scrwfile}
\TOCclone[vue d'ensemble des grandes lignes ]{toc}{stoc}
```

Cela crée un nouveau répertoire avec la rubrique « vue d'ensemble » qui est d'abord appliquée. Le nouveau répertoire utilise l'extension de fichier `stoc`. Toutes les entrées dans le fichier avec l'extension `toc` devront également être copiées dans ce répertoire. Pour activer ce nouveau répertoire qui retourne désormais seulement le niveau de chapitre, nous utilisons :

```
\addtocontents{stoc}{\protect\value{tocdepth}=0}
```

Normalement une entrée dans un répertoire peut être prise avant, jusqu'au lancement de `\begin{document}`, cela fonctionne déjà en chargeant `scrwfile` dans le préambule du document.

Représenté ici de façon non conventionnelle, `tocdepth` change le compteur dans le fichier de répertoire, changement qui reste en vigueur uniquement pour ce répertoire. Plus tard dans le document, la liste d'entrées sera alors `stoc`, gérée par `\listofstoc`, ne montrant que parties et sections du document, ce qui est assez difficile lorsqu'un aperçu de la table des matières doit être affiché.

Nous désirons créer une nouvelle table des matières avec la rubrique « Short Contents ». La nouvelle table des matières utilise un fichier d'aide avec une extension `stoc`. Toutes les entrées dans le fichier d'aide avec l'extension `toc` seront également copiées dans ce nouveau fichier d'aide. La nouvelle table des matières `short` ne doit pas avoir d'entrée de chapitre.

Ce peut être fait en utilisant :

```
\addtocontents{toc}{%
\protect\addcontentsline
{stoc}{chapter}{\protect\contentsname}%
}
```

Cependant, toutes les entrées sont copiées dans `toc` après `stoc`, et seraient donc également acceptées par l'aperçu du format de cette entrée. Donc, cet article ne peut pas être généré à partir du fichier répertoire. Depuis le paq `tocbasic` est utilisé, et :

```
\BeforeStartingTOC[toc]{%
\addcontentsline{stoc}{chapter}
{\protect\contentsname}}
```

peut être employé. Bien sûr, cela suppose que le fichier avec l'extension `toc` est également sous le contrôle de `tocbasic`. Ceci est valable pour toutes les classes de KOMA-Script. Pour plus de détails sur la commande `\AfterStartingTOC` voir la section 14.2.

Pour afficher les nouveaux contenus courts des fichiers aide de l'extension `stoc` nous allons utiliser `\listofstoc` quelque part après `\begin{document}`.

Si nous voulons aussi une entrée pour la table des matières des contenus courts, nous ne pouvons pas utiliser `\addtocontents{toc}` (écriture du contenu), `\protect\addcontentslinedefault{stoc}` (écriture du contenu court), pas plus que `\chapter` (entrée de chapitre avec `\contentsname{matières}`) car la

commande `\addcontentsline` serait trop copiée dans `stoc`. Ainsi, nous ne l'ajouterons pas au fichier `toc` et le paq `tocbasic` sera utilisé.

## 13.4 Note sur le niveau de développement

Le paq a déjà été testé par plusieurs utilisateurs et il est en cours d'utilisation, mais son développement n'est pas encore terminé. Par conséquent, il est possible théoriquement que des modifications lui soient apportées, notamment dans son fonctionnement interne, qui contient déjà une partie du code prévu pour les extensions. Aucun usager n'ayant fait de demande en ce sens, il n'existe pas actuellement de documentation utilisateur pour cela.

## 13.5 Options connues de paq incompatibles

Comme mentionné dans le chapitre 14.1, `scrwfile` redéfinit des instructions du noyau `LATEX`. Cela se produit non seulement pendant le chargement du paq, mais beaucoup plus à divers moments au cours de l'exécution d'un document, par exemple, avant de lire le fichier `aux`. Cela signifie que `scrwfile` n'est pas compatible avec les autres paqs qui redéfinissent également ces instructions lors de l'exécution.

Un exemple d'une telle incompatibilité est `titletoc`. L'exécution de ce paq est définie, dans certaines circonstances, par `\@writefile`. Utiliser `scrwfile` et `titletoc` ensemble et la fonction des deux paqs n'est plus garantie. Cette erreur de `titletoc` ne se produit qu'avec `scrwfile`.

# Chapitre 14

## Gérer les répertoires en utilisant `tocbasic`

Le but principal du paq `tocbasic` est de fournir des fonctionnalités pour les auteurs de paqs et classes afin de créer leurs propres tables ou listes de contenu tels que la liste des figures et la liste des tables et de leur permettre ainsi certains types de contrôles de répertoires sur d'autres classes ou paqs. Ce paq fournit `tocbasic` ainsi que `babel` (voir [BB13]) qui gère la langue à l'intérieur de toutes les tables et listes de contenu. Comme effet secondaire, le paq peut aussi être utilisé pour définir de nouveaux flottants ou des environnements similaires, faire des ajustements à d'autres paqs ou aux classes elles-mêmes. Pour plus de détails, voir les instructions dans le chapitre 15.2, illustrées par un exemple dans l'article 15.4, qui est repris sous une forme compacte chapitre 15.5.

Pour la table des matières et les répertoires des figures et tableaux, `tocbasic` est utilisé par KOMA-Script.

### 14.1 Commandes de base

Les instructions de base sont principalement utilisées pour gérer une liste d'extensions de fichiers connus, disponibles pour les répertoires. Les entrées dans les fichiers avec ces extensions sont généralement `\addtocontents` ou `\addxcontentsline`.

En outre, des déclarations permettent d'effectuer des actions sur toutes les extensions de fichiers. Il existe également des instructions pour effectuer les réglages d'un fichier qui appartient à une extension donnée. Typiquement une extension de fichier a aussi un propriétaire qui peut être une classe, un paq ou un identifiant qui a lui-même choisi l'auteur de la classe ou du paq que *tocbasic* utilise librement. Même *KOMA-Script* peut être, par exemple, désigné comme propriétaire des flottants utilisés pour les extensions de fichier *lof* pour les figures et *lot* pour les tables.

```
\ifattoclist{extension de fichier}{partie exécutée}{sinon else}
```

L'instruction suivante vérifie si l'extension de fichier est déjà présente dans la liste des extensions de fichiers. Si l'extension de fichier est déjà connue de cette liste, la commande est alors mise à exécution, sinon (*else*), c'est la branche *else* qui est exécutée.

Vous voulez savoir si l'extension *foo* est déjà utilisée pour afficher un message d'erreur, et si vous pouvez l'utiliser ou non :

```
\Ifattoclist{foo}{ %
\PackageError{bar}{ %
l'extension d'erreur « foo » est déjà utilisée %
}{ %
chaque extension ne peut être utilisée qu'une seule fois.
la classe ou un autre paq utilise déjà
l'extension 'foo'.
cette erreur est fatale!
vous ne devriez pas continuer!} %
}{ %
\PackageInfo{bar}{en utilisant l'extension 'foo'} %
}
```

```
\addtotoclist [propriétaire]{extension de fichier}
```

Cette instruction ajoute l'extension de fichier à la liste des extensions connues. Si l'extension de fichier est déjà enregistrée, une erreur est signalée pour éviter une double utilisation de la même extension de fichier. Si l'argument optionnel [*propriétaire*] a été spécifié, le propriétaire fixé est stocké avec cette extension de fichier.

Si l'argument optionnel est omis, *tocbasic* est testé et conservé en tant que fichier propriétaire du nom de fichier de la classe ou du paq en cours. Cela ne fonctionne que si `\addtotoclist` est appelé lors du chargement de la classe ou

du paq, mais ne fonctionne pas quand `\addtotoclist` est appelé plus tard en raison de l'utilisation d'une déclaration par l'utilisateur qui est, dans ce cas, désigné « le propriétaire ». Noter que le fichier propriétaire d'argument vide n'est pas le même que celui de l'omission complète de l'argument optionnel y compris les crochets. Un argument vide se traduirait par un propriétaire vide. Si vous voulez l'extension de fichier `foo` pour l'ajouter à la liste des extensions de fichiers connus pendant que votre paq avec le nom de fichier `bar.sty` est chargé :

```
\addtotoclist{foo}
```

Cela va ajouter l'extension de fichier `foo` avec le propriétaire `bar.sty` à la liste des extensions connues de fichiers si elle ne l'est pas déjà. Si un compte extension de fichier existe avec la classe utilisée ou un autre paq, vous obtenez l'erreur :

```
paq tocbasic Error : file extension « foo » cannot be used
twice. See the tocbasic paq documentation for explanation.
```

```
Type H <return> for immediate help.
```

```
(erreur paq tocbasic : l'extension de fichier foo ne peut
pas être utilisée deux fois. Voir la documentation du paq
tocbasic pour l'explication).
```

```
Taper H <retour> pour une aide immédiate.
```

Si vous appuyez sur « H » suivi de la touche « retour » pour recevoir l'aide, vous obtenez :

```
- File extension « foo » is already used by a toc-file, whi-
le bar.sty tried to use it again for a toc-file.
```

```
This may be eiter an incompatibility of packages, an error
at a )touche de retour) package,
```

```
or a mistake by the user -
```

```
(l'extension de fichier foo est déjà utilisée par un fichier
toc alors que bar.sty essaye de l'utiliser à nouveau.
```

```
L'erreur est due à l'incompatibilité des paqs ou à une
erreur de l'utilisateur.
```

Peut-être que votre paq fournit une déclaration préparée qui génère une liste dynamique.

Dans ce cas, utiliser l'argument optionnel de `\addtotoclist` pour préciser le fichier propriétaire :

```
\newcommand*{\createnewlistofsomething}[1]{%
\addtotoclist[bar.sty]{#1}%
```

```
\% D'autres actions pour rendre ce répertoire disponible
}
```

Maintenant, lorsque l'utilisateur appelle cette commande, par exemple avec :

```
\createnewlistofsomething{foo},
```

l'extension de fichier `foo` est ajoutée dans le fichier propriétaire `bar.sty` à la liste des extensions connues ou alors une erreur est signalée si l'extension est déjà utilisée. Vous pouvez spécifier, comme propriétaire, qui vous voulez mais il doit être unique ! Si vous êtes, par exemple, l'auteur d'un ensemble de flottants, vous pouvez spécifier que vous en êtes aussi le propriétaire au lieu de `float.sty`. Dans ce cas, les options de **KOMA-Script** pour la liste des figures et la liste des tables devraient aussi affecter les répertoires ajoutés lors de l'utilisation de l'option de la liste des extensions connues de fichiers. En effet, **KOMA-Script** enregistre les extensions de fichier `lof` pour la table des figures et `lot` pour la liste des tables avec le fichier propriétaire `float` et les options pour ce propriétaire.

```
\AtAddToTocList[le fichier propriétaire]{instructions}
```

Il est possible, de cette manière, d'ajouter à la liste interne des instructions qui seront toujours exécutées lors d'une extension avec le fichier propriétaire et les instructions ajoutées à la liste des extensions de fichiers connus. En ce qui concerne la procédure décrite, l'argument optionnel est dans l'état de `\adddtoclist`. S'il est vide, les actions sont exécutées indépendamment du propriétaire, et l'extension du fichier ajoutée à la liste des extensions connues dans ce cas. L'extension de fichier `\@currentx` est ajoutée pendant l'exécution des commandes.

Exemple : `tocbasic` utilise lui-même :

```
\AtAddToTocList []{%
  \expandafter\tocbasic@extend@babel
  \expandafter{\@currentx}}
```

chaque extension de fichier est à ajouter au répertoire existant des extensions de `tocbasic` pour le paq de `babel`.

L'utilisation deux fois de `\expandafter` dans l'exemple est nécessaire parce que l'argument de `\tocbasic@extend@babel` doit absolument être étendu. Voir aussi l'explication à propos de `\tocbasic@extend@babel`, section 15.3.

```
\removefromtoclist[fichier propriétaire]{extension}
```

Cette commande supprime l'extension de la liste des extensions connues.



Si l'argument optionnel [fichier propriétaire] est spécifié, l'extension de fichier n'est supprimée que si elle a été enregistrée.

La déclaration se réfère à `\addtotoclist` si le propriétaire est déterminé en omettant l'argument optionnel. Si un des propriétaires spécifiés est vide, aucun test de propriétaire n'aura lieu mais l'extension de fichier est supprimée indépendamment du propriétaire.

```
\doforeachtocfile[fichier propriétaire]{instructions}
```

Jusqu'à présent, vous avez appris à connaître les commandes, qui se traduisent par plus de sécurité dans le traitement des extensions de fichiers et nécessitent aussi le besoin de certains efforts supplémentaires.

Avec `\doforeachtocfile` vous allez enfin récolter le fruit de votre travail.

La commande fournit aux processus des instructions pour chaque extension de fichier connu du propriétaire donné. Pendant l'exécution des commandes `\@currentx` est l'extension du fichier en cours de traitement.

Si l'argument optionnel [fichier propriétaire] est omis, toutes les extensions de fichiers connus et indépendants du propriétaire seront utilisées. Si l'argument optionnel est vide, seules les extensions de fichier avec un propriétaire vide seront traitées.

Par exemple, si vous voulez afficher une liste de toutes les extensions connues dans le fichier journal, vous pouvez simplement écrire :

```
\doforeachtocfile{\typeout{\@currentx}}
```

Voulez-vous en revanche, que seules les extensions de fichier du propriétaire `foo`, soient classifiées :

```
\doforeachtocfile[foo]{\typeout{\@currentx}}
```

Les classes de **KOMA-Script**, `scrbook` et `scrreprt`, utilisent cette déclaration dans chaque répertoire pour lequel la propriété `chapteratlist` est spécifiée, par la façon dont vous entrez une distance verticale ou le titre du chapitre dans ce répertoire. Comment définir cette propriété est décrit au paragraphe 14.2.

```
\tocbasicautomode
```

Cette commande définit la macro `\@starttoc` de **L<sup>A</sup>T<sub>E</sub>X**.

Elle permet d'ajouter l'extension du dossier spécifié à la liste des extensions de fichiers connus, à condition qu'elle ne soit pas déjà là.

En outre, `\tocbasic@starttoc` est utilisé au lieu de `\@starttoc`. Consulter pour plus de détails au sujet de `\tocbasic@starttoc` et `\@starttoc`, l'article 15.3.

Avec `\tocbasicautomode`, chaque table des matières ou répertoire qui sera généré en utilisant `\@starttoc` sera au moins partiellement sous le contrôle de `tocbasic`. Obtenir le résultat souhaité dépend de l'extension de contrôle du paq `babel` pour toutes les tables des matières et des répertoires. Néanmoins, il est préférable que l'auteur de la classe ou du paq correspondant se serve explicitement de `tocbasic`. Il peut alors utiliser les avantages supplémentaires fournis par le paq et décrits dans les sections suivantes.

## 14.2 Génération d'une table des matières ou d'un répertoire

À la section précédente, vous avez vu des commandes permettant de gérer une liste d'extensions de fichiers connus et d'ajouter de nouvelles extensions de cette liste. Vous avez également vu une déclaration avec laquelle vous pouvez exécuter des instructions pour toutes les extensions connues d'un fichier propriétaire.

Les commandes pour gérer le fichier correspondant à une extension ou à la liste des extensions connues sont présentées dans cette section.

`\addtoeachtocfile[propriétaire]{contenu}`

La commande `\addtoeachtocfile` écrit, en utilisant `\addtocontents`, le contenu du noyau  $\text{\LaTeX}$  dans chaque fichier, qui se trouve dans la liste connue des extensions de fichiers pour le propriétaire spécifié.

Si vous omettez le contenu de l'argument optionnel, il écrit dans les fichiers de chaque extension connue de fichier. Le nom du fichier réel est dans ce cas construit à partir de `\jobname` et de l'extension de fichier. Pendant la rédaction du contenu, `\@currentx` est l'extension du fichier traité actuellement.

Vous voulez ajouter un espace vertical d'une ligne de texte dans tous les fichiers de liste des extensions de fichiers connus :

```
\addtoeachtocfile{%  
  \protect\addvspace{\protect\baselineskip}\%  
}%
```

Et si vous voulez le faire, seulement, pour les fichiers toc de foo :

```
\addtoeachtocfile[foo]{%
  \protect\addvspace{\protect\baselineskip}%
}
```

Ces instructions de service, qui ne devraient pas être déjà développées lors de l'écriture, sont de nature à être protégées dans `\addtocontents` avec `\protect`.

```
\addxcontentsline{extension}{niveau}[numéro de section]{contenu}
```

Cette déclaration est similaire à la commande `\addcontentsline` à partir du noyau de L<sup>A</sup>T<sub>E</sub>X. Cependant, elle a un argument optionnel supplémentaire pour le numéro d'entrée de la `section`, alors que ce dernier est spécifié, avec `\addcontentsline`, dans le contenu. La déclaration est utilisée pour contenir des entrées numérotées ou non dans l'extension de fichier du répertoire spécifié. Ce niveau est le nom symbolique du niveau hiérarchique et le contenu de l'entrée correspondante. Le numéro de page est automatiquement déterminé.

La déclaration elle-même teste d'abord si une commande « ajouter un niveau d'entrée de l'extension de fichier `\addEbeneaddniveau` » est définie. Si c'est le cas, elle est utilisée pour l'entrée, avec le numéro de `section` donné comme argument optionnel et passée dans le contenu comme argument obligatoire.

Un exemple d'une telle instruction fournie par les classes KOMA-Script, serait `\addparttocentry` (voir la section 21.3).

Si la déclaration correspondante n'est pas définie, on utilise à la place la commande interne `\tocbasic@addxcontentsline` qui reçoit les quatre propositions pour arguments obligatoires, puis `\addcontentsline` manuellement pour faire l'entrée souhaitée.

Pour plus de détails au sujet de `\tocbasic@addxcontentsline`, consulter la section 15.3.

L'avantage d'utiliser `\addxcontentsline` à la place de `\addcontentsline` est que le numéro de ligne de la propriété est conservé.

En second lieu, la forme des entrées, quant à la définition de critères spécifiques pour les extensions de niveaux et de fichiers peut être configurée.

```
\addxcontentslinetoeachtocfile[propriétaire]{niveau}{numéro de
classification}{contenu}
\addcontentslinetoeachtocfile[propriétaire]{niveau}{contenu}
```

Ces deux déclarations sont en relation directe avec ce qui précède déclaré dans `\addxcontentsline` ou défini dans le noyau de L<sup>A</sup>T<sub>E</sub>X `\addcontentsline`.

La différence est que ces déclarations ne se limitent pas à un seul fichier, mais agissent dans tous les fichiers propriétaires et que le renoncement du premier argument optionnel dans les fichiers de la liste spécifiée écrit dans tous les fichiers de la liste des extensions connues.

La commande `\addxcontentsline` ajoute une entrée de niveau avec extension donnée à `tocfile`. Si numéro est vide ou omis, l'entrée n'aura pas de numéro d'entrée avec le texte donné. Les entrées sans nombre peuvent être alignées à gauche au nombre des rubriques numérotées de même niveau ou en retrait comme le texte des rubriques numérotées de même niveau, selon la fonction de chaîne numérique.

Exemple : vous êtes auteur de classe et voulez écrire l'entrée de chapitre non seulement dans le fichier `toc` de la table des matières, mais aussi dans tous les fichiers `toc`, alors que « #1 » est le titre qui devrait être écrit dans les fichiers. Ce peut être fait en utilisant :

```
\addxcontentslinetoeachtocfile{chapter}
      [\thechapter]{#1}
```

Dans ce cas, bien sûr, le numéro de chapitre en cours est écrit directement dans le fichier de répertoire, s'il n'était pas protégé par `\protect` avant expansion. Alors que la rédaction de contenu est ici comme avec `\addtoeachtocfile`, `\@currentx` gère l'extension de fichier dans lequel il est écrit présentement.

Comme vous pouvez le voir, il vous suffit de remplacer `\addcontentsline` par `\addxcontentsline` pour soutenir la fonctionnalité chaîne numérique de `tocbasic`.

Noter que `\addxcontentsline` utilise `\addLevel` comme entrée de poste si une telle macro existe et sinon `\tocbasic@addxcontentsline`. Par conséquent, vous ne pouvez pas définir une macro d'entrée d'extension `\addLevel` à l'aide de `\addxcontentsline` mais avec `\tocbasic@addxcontentsline`.

La commande `\addxcontentslinetoeachtocfile`, dans la mesure où c'est possible, prend la priorité sur `\addcontentslinetoeachtocfile` puisque les extensions `\addxcontentsline` sont simplement à appliquer.

Il est préférable d'utiliser `\addxcontentsline` au lieu de `\addcontentsline` chaque fois que l'on peut.

Plus d'informations sur ces améliorations et avantages peuvent être trouvées dans l'explication précédente de la commande `\addxcontentsline`.

```

\listoftoc[titre]{extension de fichier}
\listoftoc*{extension de fichier}
\listofeachtoc[propriétaire]
\listof(nomdesextensionsdefichier)

```

Ces instructions permettent de créer les répertoires.

La version étoilée `\listoftoc*` ne nécessite qu'un seul argument, l'extension de fichier du dossier contenant les données dans le répertoire. Cette première déclaration définit les distances verticales et horizontales s'appliquant dans les répertoires, lance les instructions exécutées avant la lecture du fichier, puis lit le fichier et effectue la fin des déclarations qui sont à réaliser après cette lecture. Ce `\listoftoc*` peut être considéré comme un remplacement direct de la commande `\starttoc` du noyau de  $\text{\LaTeX}$ .

La version de `\listoftoc` sans étoiles définit l'ensemble du dossier avec le titre, table des matières optionnelle, entrée facultative dans cette table et titres courants des rubriques. Si le titre d'un seul argument est vide, une chaîne vide est utilisée. Si, toutefois, l'argument entier est omis, y compris les crochets, la commande `\listof(nomdesextensionsdefichier)` est utilisée si elle est définie. Si ce n'est pas le cas, un nom de remplacement par défaut est utilisé et un avertissement émis.

La commande `\listofeachtoc` renvoie tous les sous-répertoires propriétaires ou de n'importe quel répertoire de toutes les extensions de noms de fichiers connus. Enfin la commande `\listof'nomdesextensionsdefichier)` est définie de sorte que le titre correct puisse être délivré.

Puisque l'utilisateur lui-même pourrait éventuellement utiliser `\listoftoc` ou encore `\listofeachtoc` sans argument optionnel, il est recommandé de définir d'une manière plus appropriée `\listof(nomdesextensionsdefichier)`.

Supposons que vous avez un nouveau « répertoire des algorithmes » avec l'extension de fichier « loa » et que vous en voulez l'affichage :

```
\listoftoc[liste des algorithmes]{loa}
```

le fait pour vous, et vous pensez qu'un répertoire sans titre suffirait :

```
\listoftoc*{loa}.
```

Dans le second cas, bien sûr, une entrée facultative activée ne serait pas placée dans la table des matières. Pour plus de détails sur la propriété de l'entrée dans la table des matières voir la commande `\setuptoc`.

Si auparavant :

`\newcommand*\listofloaname}{%`  
`répertoire des algorithmes%`

ne sont définis d'une manière satisfaisante,

`\listoftoc{loa}`

sera utilisé pour créer un répertoire avec le titre désiré. Pour l'utilisateur, il est peut-être plus facile de mémoriser cette définition :

`\newcommand*\listofalgorithms{\listoftoc{loa}}`

comme déclaration de répertoire simple.

Depuis ce même répertoire, L<sup>A</sup>T<sub>E</sub>X ouvre un nouveau fichier pour l'écriture de sortie et peut appeler n'importe laquelle de ces déclarations pour un message d'erreur de style :

```
! Pas de place pour une nouvelle écriture.
\Ch@ck...\else \errmessage {Pas de place pour un nouveau #3}
\fi
```

utilisé lorsque aucun fichier d'écriture n'est disponible.

Remède dans ce cas : charger le paq `scrwfile` décrit dans le chapitre 14.

```
\BeforeStartingTOC[Extension de fichier]{instructions}
\AfterStartingTOC[Extension de fichier]{instructions}
```

Parfois, il est utile de vérifier si les états peuvent être directement exécutés avant de lire le fichier avec les instructions de données.

La macro `\BeforeStartingTOC` permet simplement de lire ces instructions, soit dans une extension de fichier unique ou dans tous les fichiers en utilisant `\listoftoc*`, `\listoftoc` ou `\listofeachtoc`.

Vous pouvez également réaliser des instructions après avoir lu, lorsque vous le définissez avec `\AfterStartingTOC`, le fichier à exécuter. Pendant l'exécution des commandes, `\@currext` est l'extension de fichier du répertoire qui est lu ou qui vient d'être lu. Un exemple d'utilisation de `\AfterStartingTOC` peut être trouvé dans la section 14.3.

```
\BeforeTOCHead[extension de fichier]{instructions}
\AfterTOCHead[extension de fichier]{instructions}
```

Si vous utilisez `\listoftoc` ou `\listofeachtoc`, ces instructions peuvent également être prédéfinies pour exécution immédiatement avant ou après le réglage de la légende. En ce qui concerne l'argument optionnel et la signification

de `\@currentx`, ce qui a déjà été expliqué ci-dessus dans `\BeforeStartingTOC` et `\AfterStartingTOC` s'applique.

`\MakeMarkcase`

Chaque fois que `tocbasic` met une marque sur un titre de colonne, il le fait comme argument de la commande `\MakeMarkcase`. Cette déclaration vise à modifier si nécessaire majuscules /minuscules du titre de la colonne. Par défaut, elle utilise une classe KOMA-Script `\@firstofone`, de sorte que l'argument lui-même reste inchangé. Si vous utilisez une autre classe `\MakeMarkcase`, sera, en revanche, en opposition avec `\MakeUppercase`. Si elle n'est pas déjà définie, l'instruction n'est fixée que par `tocbasic`. Ainsi, vous pouvez prédéfinir une classe de la manière souhaitée : elle ne sera pas redéfinie, contrairement aux habitudes, par `tocbasic`.

Si vous voulez, pour une raison étrange, que les titres courants soient affichés dans votre classe en minuscules, ce qui sera également vrai pour les titres en cours d'exécution, fixés par `tocbasic` :

```
\let\MakeMarkcase\MakeLowercase
```

Permettez-moi une remarque à propos de `\MakeUppercase` : tout d'abord, cette déclaration n'est pas entièrement extensible, ce qui signifie qu'elle peut conduire à des problèmes avec d'autres déclarations en interaction. En outre, tous les typographes conviennent que chaque phrase, mise en forme de mots ou passage en lettres majuscules nécessite une révision de l'espacement des lettres.

Pourtant, aucune distance fixe ne doit être utilisée, mais une autre distance doit être comprise entre les différentes lettres qui exigent des espaces différentes entre elles, pour composer différentes combinaisons se comportant de différentes manières. Dans le même temps, certaines formes de lettres, qui exigent une prise en compte, provoquent des trous dans le texte. Des paqs tels que `ulem` ou `soul` peuvent en faire plus que la commande `\MakeUppercase` elle-même. L'utilisation du paq de micro-caractères pour un espacement automatique est une étape, non satisfaisante, vers une solution qui n'a rien d'idéale car elle ne peut ni reconnaître, ni prendre en compte les glyphes des lettres. Cette mise en forme implique presque toujours un travail manuel qui reste plutôt réservé aux experts et nous recommandons aux amateurs pour le moins d'agir avec prudence et même, dans l'absolu, de s'abstenir et de ne pas l'utiliser en position de premier plan comme dans les rubriques

Il est important de rappeler qu'avec `hyperref`, les utilisateurs sont obligés de spécifier l'option `listname` sinon un message d'erreur signale que `hyperref` ne peut pas utiliser `\MakeUppercase` qui est pourtant nécessaire pour convertir, en majuscule, la valeur de la première lettre du type de nom d'un répertoire.

```
\defptocheading{extension}{définition}
```

Le paq `tocbasic` comprend une définition standard des caractéristiques des entêtes de répertoires, configurables comme décrit au `\setuptoc` suivant.

Mais si ces fonctionnalités ne sont pas suffisantes, une commande alternative peut être définie en utilisant `\defptocheading`, déclaration de titre de remplacement pour un répertoire avec une extension de fichier particulier. Elle peut contenir « #1 » comme seul paramètre. Lors de l'appel de la déclaration dans `\listoftoc` ou `\listofeachtoc` le titre est alors passé comme argument pour le répertoire.

```
\setuptoc{extension}{liste des Propriétés}  
\unsettoc{extension}{liste des propriétés}
```

Avec ces deux déclarations de propriétés, une extension de répertoire ou de fichier qui lui appartient peut être réglée. La liste des attributs est une chaîne de propriétés séparées par des virgules. Le paq `tocbasic` exploite les propriétés suivantes :

`leveldown` n'utilise pas le niveau supérieur de rubrique `\part` - le cas échéant `\chapter` ou `\section` si disponible - mais le premier niveau au-dessous. Cette fonctionnalité sera estimée par la commande de rubrique en interne. Mais si un utilisateur spécifie la position commande avec `\defptocheading`, il accepte la responsabilité du résultat conséquent de la fonctionnalité. `KOMA-Script` définit cette fonctionnalité en utilisant l'option `listof=leveldown` pour les extensions de fichiers du propriétaire `float`.

`nobabel` signifie que l'extension généralement utilisée automatiquement pour le changement de langue avec `babel` n'est pas utilisée pour cette extension de fichier. Cette propriété ne doit être utilisée que pour les répertoires qui sont créés dans une langue dans laquelle le changement de langue ne devraient pas être inclus dans le document. On la trouve également dans le paq `scrwfile` dont les extensions clonage sont déjà prises en charge elles-mêmes, à partir de la source de clonage. Il convient de noter que la propriété doit être définie avant d'ajouter l'extension de fichier à la liste des extensions de fichiers connus pour qu'il ait un effet.



Appliquer `noparskipfake` exclut l'utilisation d'un `\parskip` supplémentaire avant de passer `\parskip` en off. Cela signifie que, dans les paragraphes, l'espace entre le titre et la première entrée du répertoire est inférieur à la distance entre les rubriques et le texte normal, cette propriété permet un formatage plus cohérent.

`noprotrusion` empêche la désactivation de la correction de bord optique dans les répertoires. L'alignement optique des marges est désactivé par défaut lorsque le paq type de micro ou un paq qui fournit la configuration de la commande `\microtypesetup` est chargé. Ainsi, lorsque l'alignement optique est souhaité dans les répertoires, cette propriété doit être activée. Il convient toutefois de noter que la compensation optique périphérique peut être mal réglée et donner de mauvais résultats.

`numbered` utilise un titre numéroté pour la table des matières ou toute autre liste et génère également une entrée dans cette même table des matières. Cette fonctionnalité donne au titre une valeur de numéro d'écriture par une instruction interne. D'autre part, si un utilisateur définit lui-même la rubrique commande avec `\deftocheading`, cet utilisateur est responsable du résultat. Les classes de `KOMA-Script` définissent cette fonctionnalité en utilisant l'option `listof=numbered` pour toutes les extensions de fichiers de l'intitulé du flottant.

`numberline` cite le numéro de ligne d'article écrit avec `\addxcontentsline` ou avec `\addxcontentslinetoc` dans le répertoire, où l'argument du nombre manquant ou vide doit comprendre une déclaration de `\numberline` (numéro de ligne) qui peut être vide. Cela entraîne le fait que ces entrées ne sont pas définies alignées à gauche avec le nombre, mais sont réglées au même niveau que le texte des entrées numérotées. Les classes de `KOMA-Script` définissent cette propriété avec `listof=numberline` pour les extensions de fichier propriétaire flottant et `toc=numberline` pour l'extension de fichier. En conséquence, lorsque vous utilisez l'option `listof=nonnumberline` ou l'option `toc=nonnumberline`, cette propriété est à chaque fois remise à zéro .

`onecolumn` signifie que le mode colonne  $\LaTeX$  utilise automatiquement pour ce répertoire `\onecolumn`. Cela s'applique uniquement si ce répertoire n'est pas déplacé avec la propriété décrite ci-dessus à un niveau hiérarchique bas. Les classes `KOMA-Script` `scrbook` et `scrreprt` définissent, pour tous les flottants liés à un identifiant, cette propriété via `\AtAddToTocList` (ou liés à eux-mêmes en tant que propriétaire). La table des matières des deux classes, liste des figures et liste des tables, est automatiquement mise en colonne. Le mode multi-colonne du paq `multicol` n'est pas reconnu ou modifié par l'option.

`totoc` ajoute le titre du répertoire à inclure dans la table des matières. Cette capacité est contrôlée par une instruction interne. Si l'utilisateur définit lui-même la fonction avec `\deftocheading`, il est responsable du résultat. Les classes `KOMA-Script` fixent cette fonctionnalité en utilisant l'option `listof=totoc` pour toutes les extensions de l'intitulé du flottant.

Les classes `KOMA-Script` utilisent aussi `chapteratlist`, une fonctionnalité qui active un code spécial mis dans la liste au début d'un nouveau chapitre.

Ce code peut être, soit un espace vertical, soit le titre du chapitre par défaut.

Voir `listof` en section 3.20 pour plus d'informations sur ces caractéristiques.

Depuis que les classes de `KOMA-Script` utilisent `tocbasic` pour la liste des figures et la liste des tables, il existe une autre façon d'empêcher la publication du sommaire de ces répertoires :

```
\unsettoc{lof}{chapteratlist}
\unsettoc{lot}{chapteratlist}
```

il suffit de définir le répertoire de fichier des chapitres à utiliser avec l'extension `loa`, si vous voulez qu'une colonne soit automatiquement positionnée dans les classes avec `\chapter` comme plus haut niveau de détail. Et si vous voulez ajouter le chapitre structuration des classes de `KOMA-Script` à votre propre liste d'algorithmes avec l'extension de fichier `loa` des exemples précédents, vous pouvez utiliser :

```
\setuptoc{loa}{chapteratlist}
```

Vous pouvez aussi, dans les classes qui utilisent `\chapter` comme plus haut niveau de détail dans le répertoire, forcer également le mode colonne pour cette liste avec ;

```
\ifundefinedorrelax{chapter}{}{%
\setuptoc{loa}{onecolumn}%
}
```

L'utilisation de `\ifundefinedorrelax` présume le paq `scrbase` (voir section 12.3). Il importe peu que le paq soit utilisé avec une autre classe. Vous ne devriez pas avoir besoin de définir cette fonctionnalité, car si l'autre classe reconnaît également la fonction de votre paq, elle l'utilisera automatiquement.

```
\iftocfeature{extension de fichier}{propriétés}{instruction}{complément}
```

Cela peut être indiqué pour chaque propriété et cette commande peut être utilisée pour tester si une fonction a été fixée pour une extension de fichier.

Si c'est le cas, alors l'instruction est exécutée, sinon c'est la partie complément (else) par défaut. Cela peut être utile, par exemple, si vous définissez vos propres déclarations d'entête avec `\defptocheading`, mais que vous voulez soutenir les propriétés décrites ci-dessus `totoc`, `numbered` ou `leveldown`.

## 14.3 Configuration des entrées de répertoire

Outre les répertoires réels et leurs fichiers d'aide, le paq `tocbasic` permet, à partir de la version 3.20, d'influencer également les entrées de répertoire et de nouveaux styles peuvent être définis, mais il y a aussi plusieurs styles prédéfinis disponibles. À moyen terme, `tocbasic` ne remplacera jamais officiellement le paq `tocstyle` de *KOMA-Script* dont les classes sont elles-mêmes construites, également depuis la version 3.20, entièrement sur les informations fournies par les styles `tocbasic` pour les entrées de répertoire.

### tocdepth

Les entrées de répertoire sont généralement organisés hiérarchiquement. Pour cela, chaque niveau d'entrée sera considéré comme une valeur numérique. Plus la valeur est grande, plus le niveau est profond dans la hiérarchie. Dans les classes `standard` les parties ont par exemple une valeur de -1 et le chapitre une valeur de 0. Le compteur  $\LaTeX$  `tocdepth` détermine à ce propos jusqu'à quel niveau les entrées sont affichées dans le répertoire.

Dans `book`, `tocdepth` présélectionne par exemple deux niveaux avec entrées dans `part`, `chapter`, `section` et `subsection`. Des niveaux plus profonds tel que `subsubsection` dont la valeur numérique est 3 ne sont pas émis. Néanmoins, les fichiers auxiliaires pour la table des matières sont disponibles. Les styles d'entrée définis par `tocbasic` considèrent à `part` `gobble` (voir `DeclareTOCStyleEntry`) ainsi que `tocdepth`.

```
\numberline{numéro de classement}
\usetocbasicnumberline[code]
```

Bien que le noyau  $\LaTeX$  définisse déjà une commande `\numberline`, ceci est cependant insuffisant pour les exigences de `tocbasic`. En conséquence, il définit ses propres instructions et `\numberline`, ensemble si nécessaire au moyen de `\usetocbasicnumberline` pour les éléments de la liste. Une redéfinition de `\numberline` est donc souvent inefficace lorsque vous utilisez `tocbasic` et peut également conduire à des avertissements.

Vous pouvez également utiliser la définition de `tocbasic` en appelant, dans le préambule du document, `\usetocbasicnumberline`. La déclaration tente d'abord de déterminer si des instructions internes importantes sont utilisées par `tocbasic` dans la définition actuelle et `\numberline` est redéfini dans ce cas pour exécuter du code supplémentaire. Mais si aucun argument n'est spécifié en option, `\PackageInfo` envoie un message, appliqué à la redéfinition, qui peut être supprimé simplement par une spécification d'un argument optionnel vide. Noter que `\usetocbasicnumberline` peut changer le commutateur interne `\@tempswa` global.

`\DeclareTOCStyleEntry`[liste d'options]{style}{niveau d'entrée}

Utiliser ces instructions aux entrées de répertoire pour un niveau particulier d'entrée déclarée ou configurée.

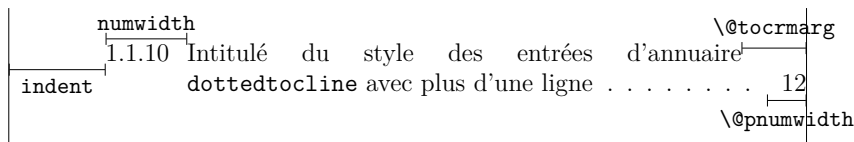


FIGURE 14.1 – Illustration de certains attributs de style d'entrée de répertoire de `dottedtocline`

Voici le niveau d'entrée du nom de niveau respectif d'entrée, pour la `section` exemple, appartenant au même niveau hiérarchique d'entrée TOC, ou de la figure pour l'entrée d'une figure dans le tableau des figures. Chaque niveau d'entrée est associé à un style particulier qui doit déjà être défini au moment de la déclaration. La liste des différentes options, la plupart du temps dépendantes, définit les propriétés de style de l'élément qui peuvent être réglées. Actuellement les styles d'entrée définis de `tocbasic` sont les suivants :

Le style par défaut est un clone de `dottedtocline`. Pour les écrivains des classes qui utilisent `tocbasic`, il est recommandé de changer ce style en utilisant `\CloneTOCStyle` dans la liste de style de répertoire par défaut de la classe. Dans les classes `KOMA-Script`, ce sera par défaut un clone de style `tocline`. `dottedtocline` correspondant au style standard des classes `book` et `report` pour les niveaux d'entrées du répertoire, de `section` à `subparagraph` et reconnu de toutes les classes `standard` pour les entrées de la liste des figures et des tables.

Ce style ne connaît que le niveau des caractéristiques, `indent` et `numwidth`. Les entrées sont en retrait de la sortie gauche, marquées par un tiret, dans la police courante et `\numberline` n'est pas redéfini, `numwidth` détermine la

largeur du nombre. Les entrées multi-lignes de la collection sont augmentées par `numwidth` à partir de la deuxième ligne. Le numéro de page est géré dans la police normale avec `\normalfont`. La saisie de texte et le numéro de la page sont reliés par une ligne pointillée (voir les propriétés du style dans la figure page ci-contre). `Gobble` est le style concevable le plus simple. Les entrées de ce style ne seront pas émises indépendamment de tous les paramètres pour `tocdepth` mais, pour ainsi dire, ingérées. Néanmoins, le niveau de propriété par défaut n'est jamais évalué.

`targetocline` correspondant au style qui fait partie du niveau des classes `standard` connues. Il ne connaît que les propriétés `niveau` et `tiret`. Ce dernier est, en quelque sorte, une dérogation aux classes `standard` qui ne prennent en charge aucune collecte de message partiel. Un saut de page est facilité avant l'inscription et les entrées, indentées à gauche, sortent en `\large\bfseries`. Si `\numberline` est utilisée, la largeur du nombre est fixée à `3em` et `\numberline` n'est pas redéfinie. Utiliser les classes `standard` pour les messages partiels sans `\numberline`. La valeur `n` a aucun effet sur la collecte de la deuxième ligne dans les entrées multi-lignes.

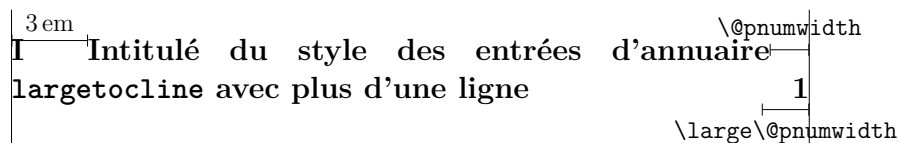


FIGURE 14.2 – Illustration de certains attributs de style d'entrée de répertoire de `targetocline`

Les propriétés du style sont illustrées dans la figure 14.2. Il est également frappant de constater que le style a pris un certain nombre d'incohérences dans la classe `standard`, par exemple, l'absence de collecte de la deuxième ligne dans les entrées multi-lignes et deux valeurs différentes de `\@pnumwidth` résultant d'une fonction de la taille de la police. Il en résulte aussi du fait que, dans un cas extrême, le texte de la rubrique du numéro de page peut venir trop près. Il convient de noter que la largeur du numéro de la `section` représentée sur la figure est utilisée seule, malgré l'usage de `\numberline`. Les classes `standard` définissent cependant le nombre avec une distance fixe de `1em`.

`tocline` est un style souple utilisé par défaut pour toutes les entrées des classes `KOMA-Script`. En conséquence, ces classes définissent également les clones `part`, `chapter` et `section`, respectivement `section` et `subsection`, en utili-

sant ce style, puis changent le code d'initialisation des clones de sorte qu'ils ont des préférences différentes.

Le style connaît le niveau des propriétés, `beforekip`, `dynnumwidth`, et de `entryformat`, `entrynumberformat`, `breakafternumber`, `indent`, `linefill`, et `numsep`, `numwidth`, `onstarthigherlevel`, `onstartlowerlevel`, et encore de `onstartsamelevel`, `pagenumberbox`, `pagenumberformat` et `raggedpagenumber`. Les préférences de toutes ces propriétés sont déterminées en fonction du nom du niveau d'entrée et orientées sur la base des résultats des classes `standard`. Il est donc possible de changer, après le chargement `tocbasic`, le style de la table des entrées de contenu dans la classe `standard` avec `\DeclareTOCEntryStyle` dans `tocline` sans conduire à des changements majeurs dans l'apparition du contenu de la table des entrées.

Ils sont applicables aux propriétés nécessaires aux changements souhaités. La même chose est vraie pour les figures et les tables des classes `standard`.

En raison de sa grande flexibilité, ce style peut remplacer, en principe, les styles `dottedtocline`, `toc!undottedtocline` et `largetocline` mais nécessite donc en partie une difficulté accrue de trouver la configuration. La figure page ci-contre illustre quelques propriétés de longueurs du style. D'autres sont présentées dans la figure page 438.

`undottedtocline` correspondant au style utilisé par les classes `standard book` et `report` pour les niveaux connus de `chapter` et de `section`. Il ne connaît que les caractéristiques de niveau `indent` et `numwidth`. Ce dernier est, en quelque sorte, comme une exception par rapport aux classes `standard` qui n'ont aucun aménagement pour les niveaux mentionnés.

Avant que l'entrée ne soit enregistrée, un saut de page est facilité et une distance verticale ajoutée, introduite. Les entrées, en `\bfseries`, sont indentées sur le côté gauche. Le `\numberline` n'est pas redéfini. La largeur du nombre est déterminée par `numwidth`. Pour les entrées multi-lignes, la collection sera augmentée, à partir de la deuxième ligne, par `numwidth`, voir la figure page ci-contre qui illustre les propriétés du style.

Une explication des caractéristiques des styles `tocbasic` définis peut être trouvée dans la figure 14.1 page 438. Dans les propriétés utilisées comme options de `\DeclareNewTOC`, `\DeclareSectionCommand` et les instructions connexes se trouvent les noms à fournir avec le préfixe `toc`, ainsi que, par exemple `toclevel` à utiliser au lieu de `level`.

En fin de compte, l'appel à `\DeclareTOCStyleEntry` conduit à la définition du niveau d'entrée `\l@Eintragsebene`.

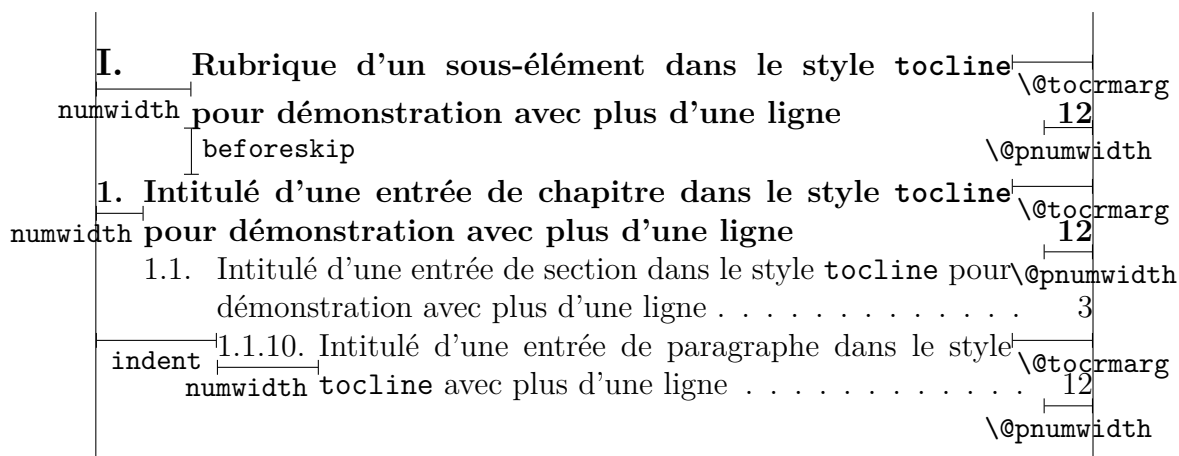


FIGURE 14.3 – Illustration de certains attributs de style d’entrée de répertoire de `tocline`

Une instructions complexe de KOMA-Script, `\DeclareTOCEntryStyle`, dédiée expressément aux développeurs de  $\text{\LaTeX}$  et non aux utilisateurs permet de définir un nouveau style pour les entrées de répertoire habituellement générées avec `\addcontentsline` ou lorsque vous utilisez `tocbasic` de préférence avec `\addxcontentsline` (voir l’article 15.1). Ici  $\text{\LaTeX}$  écrit une commande associée `\contentsline` dans le fichier auxiliaire respectif.

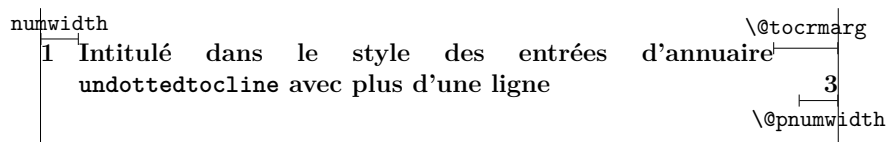


FIGURE 14.4 – Illustration de certains attributs de `undottedtocline` comme style d’entrée de répertoire pour un titre de chapitre

Ce fichier d’aide découvre une instruction  $\text{\LaTeX}$  `\l@niveaudentrée` pour chaque `\contentsline`.

Si plus tard, le style du niveau d’entrée dans `\DeclareTOCStyleEntry` est affecté, le code d’initialisation en option est d’abord exécuté et le niveau d’entrée ensuite défini par `\l@coded' instruction`. Alors le code de commande devient finalement le code qui est exécuté dans `\l@niveaudentrée`. Avec `#1`, nom du niveau d’entrée, alors que `##1` et `##2` sont des espaces réservés pour les deux arguments de `\l@niveaudentrée`.

TABLE 14.1 – Attributs pour le répertoire prédéfini liste de styles de *tocbasic*


---

<p><b>beforeskip</b>=<i>Länge</i></p> <p>Espace vertical inséré, avant une entrée de ce niveau, dans le style <code>tocline</code> (voir la figure 14.3 page précédente) en fonction du niveau <code>\vskip</code> ou <code>\addvspace</code> et compatible avec les classes <code>standard</code> et les versions antérieures de <code>KOMA-Script</code>. Au niveau de l'entrée <code>part</code>, l'attribut est initialisé avec 2.25em plus 1pt et avec <code>chapter</code> avec 1pt plus 1em. Si aucun niveau d'entrée de chapitre n'est connu, 1pt est utilisé pour la <code>section</code> également plus 1em. Les autres niveaux sont initialisés avec 0pt plus .2pt.</p>	<p>-----</p> <p><b>breakafternumber</b>=<i>interrupteur</i></p> <p>Une des valeurs pour les interrupteurs simples dans la table 1.6 page 20. Si le commutateur est activé dans le style <code>tocline</code>, une nouvelle ligne démarre avec le nombre fixé par <code>\numberline</code>, alignée à gauche et numérotée. Par défaut, la propriété n'est pas définie dans le style <code>tocline</code>. Un répertoire avec <code>\setuptoc</code> attribue un numéro de propriété de ligne, comme dans les classes <code>KOMA-Script</code> et utilise l'option <code>toc=numberline</code>. Cela signifie aussi que, pour les entrées non numérotées, la ligne est définie avec un numéro vide dans le format <code>entrynumberformat</code>.</p>
<p>-----</p> <p><b>dynnumwidth</b>=<i>interrupteur</i></p> <p>Une des valeurs pour les interrupteurs simples dans la table 1.6 page 20. Si le commutateur est activé dans le style <code>tocline</code>, la propriété <code>numwidth</code> est ignorée et <code>LATEX</code> applique alors la dernière largeur maximale des numéros du même niveau, plus la valeur de <code>numsep</code> utilisée en entrée.</p>	<p>-----</p> <p><b>entryformatformat</b><i>entryformat</i>=<i>commande</i></p> <p>Cette propriété permet de modifier la mise en forme de l'entrée. La valeur n'a qu'un seul argument non extensible en totalité alors que les commandes comme <code>\MakeUppercase</code> attendent un argument entièrement extensible et ne peuvent donc pas être utilisées à ce stade. Les changements de police sur le format d'entrée sont effectués en utilisant <code>\normalfont</code> et <code>\normalsize</code>. Il convient de noter que <code>linefill</code> et le numéro de page sont indépendants du format d'entrée. Voir aussi les propriétés de page <code>numberformat</code>. L'argument de la propriété du niveau d'entrée pour <code>part</code> est initialisé avec <code>\large\bfseries</code> et pour <code>chapter</code> avec <code>\bfseries</code>. Si lors de l'initialisation de <code>section</code> aucun niveau de chapitre n'existe, <code>bfseries</code> est également utilisé pour ce niveau. Pour tous les autres niveaux, l'argument reste inchangé.</p> <p>-----</p>



`entrynumberformat=commande`

La mise en forme est définie par `\numberline` et l'enregistrement du numéro peut être changé. La commande inclut un seul argument de valeur. Le changement de police commence avec le `format` de la propriété d'entrée. Cette propriété reprend l'édition des arguments passés et le numéro d'enregistrement sort donc inchangé. Si un répertoire avec `\setuptoc` définit la propriété de numéro de ligne (voir l'article 15.2) comme avec les classes KOMA-Script et utilise dans ce cas l'option `toc=numberline`, cela signifie aussi que, pour les entrées non numérotées, la commande est toujours en cours d'exécution.

`indent=longueur`

Distance horizontale entre l'entrée et le bord gauche (voir la figure page 438, voir la figure 14.2 page 439 et voir la figure 14.3 page 441).

Lorsque l'initialisation du style `tocline` est faite pour tous les niveaux d'entrée dont le nom commence par « sub », avec indentation + `numwidth`, utilise le même niveau d'entrée sans ce préfixe, si un tel niveau existe avec des propriétés correspondantes. Dans les styles `dottedtocline`, `undottedtocline` et `tocline`, on trouve pour le niveau d'entrée de `part` jusqu'à `subparagraph`, aussi bien dans figure que dans table, une initialisation avec des valeurs correspondant aux classes `standard`. Tous les autres niveaux ne recevront aucune initialisation. Pour eux, la première utilisation d'une indication explicite est donc obligatoire.

`level=entier`

Valeur numérique du niveau d'entrée. En effet, seules apparaissent les entrées dont le niveau numérique ne dépasse pas `tocdepth`. Cette propriété, obligatoire pour tous les styles, est automatiquement intégrée dans la définition d'un style. Le style `tocline` initialise tous les niveaux d'entrée dont le nom commence par « sub », selon une valeur plus élevée d'un niveau de la même entrée sans ce préfixe, si un tel niveau existe. Dans les styles `dottedtocline`, `largetocline`, `tocline` et `undottedtocline`, on trouve pour le niveau d'entrée de `part`, `chapitre`, `section`, `subsection`, `subsubsection`, `paragraphe`, `alinéa`, `figure` et `table` une initialisation automatique indexée sur le nom. Pour les autres niveaux, l'initialisation avec la valeur du niveau de plan est définie, si cela est compatible avec les classes KOMA-Script, par `\Eintragsebenenenumdepth` (`\profondeur du niveau d'entrée`).

`linefill=code`

Le style `tocline` peut changer entre la fin du texte d'entrée et le numéro de page, le type de remplissage. La propriété `linefill` reçoit directement le code désiré comme valeur. La propriété du niveau d'entrée pour `part` et `chapter`

est initialisée avec `\hfill`. Cela place le numéro de page sur le bord droit. Au cas où aucun niveau d'entrée de chapitre n'est défini, cela vaut aussi pour `section`. Tous les autres niveaux sont initialisés avec `\TOCLineLeaderFill`. Les codes indiqués ne conduisent pas automatiquement à un remplissage de la distance, d'ailleurs, la propriété `raggedpagenumber` doit être définie afin de ne pas provoquer un message « `underfull \hbox` ».

-----  
`numsep=longueur`

Le style `tocline` essaie de faire en sorte que, entre le nombre et le texte de l'entrée, un intervalle de longueur soit respecté. Avec `dynnumwidth` activé réservé pour la largeur de nombre, `numwidth` peut être corrigé en conséquence. S'il n'est pas activé, `dynnumwidth` peut seulement émettre un avertissement lorsque cette condition n'est pas remplie. La valeur par défaut est de `0,4em`.

-----  
`numwidth=longueur`

Pour le nombre d'une largeur d'entrée réservée (voir la figure 14.1 page 438 jusqu'à la figure 14.4). Cette valeur est ajoutée, dans les styles `dottedtocline`, `tocline` et `dottedtocline`, à partir de la deuxième ligne de l'entrée avec un retrait à gauche. Le style `tocline` initialisé pour tous les niveaux `tocentrystyle` d'entrée dont le nom commence par « `sub` », avec la valeur du niveau de la même entrée sans ce préfixe fixée à `0,9em`, est effectué si un tel niveau existe avec les propriétés appropriées. Dans les styles `dottedtocline`, `undottedtocline` et `tocline` on trouve comme niveau d'entrée de `part` jusque `subparagraph`, aussi bien dans figure que dans table, l'initialisation avec des valeurs correspondant aux classes `standard`. Tous les autres niveaux ne recevront aucune initialisation et pour eux, la première utilisation d'une indication explicite est donc obligatoire.

-----  
`onstarthigherlevel=code`

Le style `tocline` peut l'exécuter au début d'une action selon que la dernière série d'entrée avait une valeur supérieure, de même niveau ou d'un niveau inférieur. Dans le cas où l'entrée actuelle a une valeur de niveau supérieur, donc plus faible dans la hiérarchie des entrées, la propriété du code spécifié est exécutée. La détection fonctionne seulement que tant que `\lastpenalty` n'a pas changé depuis la dernière entrée. Elle est initialisée avec la propriété `\LastTOCLevelWasLower`.

-----  
`onstartlowerlevel=code`

Le style `tocline` l'exécutera au début d'une action selon la valeur supérieure, de même niveau ou de niveau inférieur de la dernière série d'entrée. Dans le cas où l'entrée actuelle a une valeur de niveau inférieur, donc plus élevé dans la hiérarchie des entrées, la propriété du code spécifié est exécutée. La détection

fonctionne seulement tant que `\lastpenalty` n'a pas changé depuis la dernière entrée. Elle est initialisée avec la propriété `\LastTOCLevelWasLower` ce qui se traduit normalement par une rupture opérée avant l'entrée.

-----  
`onstartsamelevel=code`

Le style `tocline` peut l'exécuter au début d'une action selon la valeur de la dernière série d'entrée : supérieure, de même niveau ou de niveau inférieur. Dans le cas où l'entrée actuelle a une valeur égale de niveau dans la hiérarchie des entrées, la propriété du code spécifié est exécutée. La détection fonctionne seulement tant que `\lastpenalty` n'a pas changé depuis la dernière entrée. Elle est initialisée avec la propriété `\LastTOCLevelWasLower` ce qui se traduit normalement par une rupture opérée avant l'entrée.

-----  
`pagenumberbox=commande`

Appartenant habituellement à un numéro de page d'entrée aligné à droite dans une boîte de largeur `\@pnumwidth`. Avec le style `tocline`, on peut configurer la commande qui est utilisée pour cette propriété et dans ce cas, prévoir de spécifier un seul argument qui initialise la propriété avec la boîte ci-dessus.

-----  
`pagenumberformat=commande`

Avec cette propriété, la mise en forme du numéro de page de l'entrée peut être modifiée. La commande attend un argument à prévoir. Le changement de police sur le format d'entrée a lieu à partir du format d'entrée suivi de `\normalfont \normalsize`. la propriété pour le niveau d'entrée est initialisée pour une partie du niveau de commande avec la sortie de l'argument passé dans `\large`. Pour tous les autres niveaux, la sortie est `\normalfont \normalcolor`.

-----  
`raggedentrytext=interrupteur`

Une des valeurs pour les interrupteurs simples dans la table 1.6 page 20. Si le commutateur est activé dans le style `tocline`, le texte de l'entrée ne sera pas justifié mais flottant et ne concerne que les mots qui sont sur plusieurs lignes. Cette option n'est pas définie par défaut.

-----  
`raggedpagenumber=interrupteur`

Une des valeurs pour les interrupteurs simples dans la table 1.6 page 20. Si le commutateur active le style `tocline`, le numéro de page n'est pas mis en alignement nécessairement à droite. L'activation de cette option, par la valeur du remplissage de `linefill`, peut provoquer l'apparition ou la disparition d'une alerte ou même affecter concrètement la mise en forme des messages. Il est donc important de définir ces deux propriétés en correspondance mutuelle. Par défaut, cette option est pas définie et convient pour initialiser `linefill` aussi bien avec `\hfill` qu'avec `\TOCLineLeaderFill`.

---

L'initialisation du code sert à initialiser tous les réglages. Le développeur doit veiller à ce que tous les paramètres d'une valeur figurent réellement, de sorte que `\DeclareTOCStyleEntry` fonctionne de façon correcte sans fournir une liste d'options. En outre, le code d'initialisation a également toutes les options comprenant le style particulier défini, le style prédéfini obligatoire et le style un seul niveau. La valeur définie pour le niveau peut être interrogée dans le code de commande avec `\@nameuse{#1tocdepth}` à comparer par exemple avec la valeur du compteur `tocdepth`.

```

\DeclareTOCEntryStyle{Style}[initialisation]{code de commande}
\DefineTOCEntryOption{Option}[valeur par défaut]{code}
\DefineTOCEntryBooleanOption{Option}[valeur
par défaut]{Préfixe}{Postfix}
{déclaration}
\DefineTOCEntryCommandOption{Option}[valeur
par défaut]{Préfixe}{Postfix}
{déclaration}
\DefineTOCEntryIfOption{Option}[valeur par défaut]{Préfixe}{Postfix}
{déclaration}
\DefineTOCEntryLengthOption{Option}[valeur par défaut]{Préfixe}{Postfix}
{déclaration}
\DefineTOCEntryNumberOption{Option}[valeur par défaut]{Préfixe}{Postfix}
{déclaration}

```

Définir de nouvelles options des propriétés d'un niveau d'entrée est limité dans les instructions d'initialisation :

- `\DefineTOCEntryBooleanOption`,
- `\DefineTOCEntryCommandOption`,
- `\DefineTOCEntryIfOption`,
- `\DefineTOCEntryLengthOption`
- `\DefineTOCEntryNumberOption`.

Toutes ces déclarations définissent en option la commande :

- `\PrefixniveauentréePostfix`

avec la valeur donnée ou, en l'absence d'une affectation de valeur, en fixe les limites lorsqu'elle est appelée.

Une caractéristique particulière de `\DefineTOCEntryIfOption` qui permet de définir `\PrefixniveauentréePostfix` intégrant toujours une déclaration de deux arguments comme option des valeurs d'activation dans la table 2.5 page 132, avec expansion de la déclaration pour le premier argument. Une

valeur de désactivation passée comme valeur de l'option élargit l'instruction du deuxième argument.

La Déclaration devrait être un court texte décrivant la possible signification de l'option en quelques mots clés et `tocbasic` délivre les messages d'erreur, les avertissements et des informations sur le terminal et dans le fichier journal. Le style simple de `tocbasic`, `gobble`, a été chargé et défini par :

```
\DeclareTOCEntryStyle{gobble}{}%
```

Maintenant avec :

```
\DeclareTOCStyleEntry[level=1]{gobble}{dummy}
```

est défini un niveau d'entrée `dummy` dans ce style, qui, entre autres choses, correspondrait à :

```
\def\dummytocdepth{1}
```

```
\def\l@dummy#1#2{}
```

Dans le style `tocline`, un exemple avec :

```
\DefineTOCEntryCommandOption{linefill}[\TOCLineLeaderFill]%
{scr@tso@}{@linefill}{filling between text and page number}%
```

utilisé pour définir la ligne d'option en fixant des limites avec :

```
\cott{TOCLineLeaderFill}.
```

Un appel comme :

```
\RedeclareTOCStyleEntry[linefill]{tocline}{part}
```

permet donc d'inclure la définition de la commande pour exécuter :

```
\def\scr@tso@part@linefill{\TOCLineLeaderFill}
```

Pour définir vous-même votre propre style, il est suggéré d'étudier d'abord la définition du style de `dottedtocline` qui comporte beaucoup d'indices sur la façon dont les instructions sont à utiliser avant de passer aux définitions beaucoup plus complexes du style `tocline`.

Dans de nombreux cas, il est suffisant pour cloner un des styles existants avec `\CloneTOCEntryStyle`, modifier l'initialisation avec `\TOCEntryStyleInitCode` ou `\TOCEntryStyleStartInitCode`. `\DefineTOCEntryOption` est plutôt utile pour la définition des instructions restantes et ne doit pas être exploité dans une règle directement. Normalement, ceci n'est pas une nécessité et n'est mentionné ici que pour être complet.

```
\CloneTOCEntryStyle{style}{nouveau style}
```

Cette déclaration s'applique à un style existant qui peut être cloné pour créer un nouveau style déclaré avec les mêmes caractéristiques et préférences que le style existant. Le paq lui-même utilise la valeur de style par défaut `\CloneTOCEntryStyle` pour déclarer un clone de `dottedtocline`. Les classes KOMA-Script utilisent la déclaration `style` de `part`, `section` et `chapitre` ou `paragraphe` pour déclarer comme un clone de `tocline` et un nouveau style clone par défaut pour déclarer `section` ou `subsection`.

```
\TOCEntryStyleInitCode{style}{initialisation}
\TOCEntryStyleStartInitCode{style}{initialisation}
```

Chaque entrée de répertoire a un code d'initialisation de style appelé avec `\DeclareTOCEntryStyle` quand un niveau du style de répertoire correspondant est appliqué. L'initialisation n'a pas d'effet secondaire étendu car elle est aussi utilisée localement dans d'autres commandes comme `\DeclareNewTOC`. Elle sert à définir, d'une part les propriétés pour le style particulier et d'autre part les paramètres par défaut pour ces propriétés.

L'utilisation de `\TOCEntryStyleStartInitCode` et `\TOCEntryStyleInitCode` permet de définir un code d'initialisation de style qui peut être prolongé par une nouvelle initialisation. Ici `\TOCEntryStyleStartInitCode` ajoute `ngscode` au nouveau code d'initialisation alors que `\TOCEntryStyleInitCode` favorise un retour au code existant. Les classes KOMA-Script l'exploitent, par exemple, pour initialiser une partie du remplissage appropriée, espacement vertical, style et police pour la partie de `tocline` cloné. Ainsi les classes `scrbook` et `scrreprt` utiliseront :

```
\CloneTOCEntryStyle{tocline}{section}
\TOCEntryStyleStartInitCode{section}{%
\expandafter\providecommand%
\cscname scr@tso@#1@linefill\endcscname
{\TOCLineLeaderFill\relax}%
}
```

pour définir le style `section` comme clone modifié `tocline`.

```
\LastTOCLevelWasHigher
\LastTOCLevelWasSame
\LastTOCLevelWasLower
```

Avec les entrées dans le style de `tocline` démarre `\lastpenalty` dépendant de l'exécution de l'un de ces trois états.

Ici `\LastTOCLevelWasHigher` et `\LastTOCLevelWasSame` ajoutent dans le mode vertical `\addpenalty{\@lowpenalty}` et permettent ainsi une mise en page avant les entrées de même niveau ou d'un niveau supérieur. D'autre part, `\LastTOCLevelWasLower` est défini vide jusqu'à présent et interdit donc la mise en page entre l'entrée et la première sous-entrée.

Les utilisateurs ne devraient pas redéfinir ces instructions, mais vous pouvez, et même devez, les changer pour les niveaux d'entrée respectifs créés spécifiquement avec les caractéristiques `onstartlowerlevel`, `onstartsamelevel` et `onstarthigherlevel`.

`\TOCLineLeaderFill[caractère de remplissage]`

Cette déclaration est destinée à être utilisée en tant que valeur de la propriété `linefill` du style `tocline` de la fiche répertoire. Une connexion est ensuite générée entre la fin du texte d'une entrée et le numéro de page associée. Le caractère de remplissage, qui est répété à intervalle régulier, peut être spécifié comme un argument optionnel. Par défaut, il s'agit d'un point. Comme le nom l'indique, ce caractère est fixé à l'aide de `\leaders`. Lorsque la distance est similaire à `\@dottedtocline` du noyau L<sup>A</sup>T<sub>E</sub>X avant et après le caractère de remplissage, `\mkern\@dotsep` est utilisé.

## 14.4 Commandes internes pour auteurs

Le paq `tocbasic` fournit des instructions internes pour l'utilisation partagée par les auteurs de classes et de paqs qui commencent toutes par `\tocbasic@`. Les auteurs doivent exclusivement les utiliser en l'état et ne pas chercher à les redéfinir. La fonction interne peut toujours être modifiée ou prolongée, mais une redéfinition des instructions peut endommager sérieusement le fonctionnement du paq `tocbasic`.

`\tocbasic@extend@babel{extension de fichier}`

Le paq `babel` (voir [BB13]) ou un noyau de L<sup>A</sup>T<sub>E</sub>X qui a été étendu à la gestion des langues de `babel`, écrit des instructions pour changer la langue dans les fichiers avec les extensions `toc`, `lof` et `lot` à chaque changement du langage usuel à réaliser, soit au commencement, soit à l'intérieur d'un document. `tocbasic` étend ce mécanisme de sorte que l'utilisation de `\tocbasic@extend@babel` bénéficie également à d'autres extensions de fichier.

L'extension du fichier d'argument doit alors être totalement élargie, sinon, le sens de l'argument peut changer jusqu'à son utilisation réelle. Cette commande

est normalement ajoutée, par défaut, à la liste des extensions connues de chaque fichier en utilisant `\addtotoclist`. Elle peut être supprimée en utilisant la fonction `nobabel` (voir `\setuptoc`, article 14.2).

Pour les extensions de fichiers `toc`, `lof` et `lot`, `tocbasic` le fait spontanément pour éviter que la commutation n'entre à plusieurs reprises dans les fichiers appropriés.

Il n'y a donc aucune raison, normalement, d'appeler cette commande vous-même. Cependant, il est concevable qu'il existe des répertoires que `tocbasic` ne contrôle pas, qui ne sont pas inclus dans la liste des extensions de fichiers connus, mais qui doivent, néanmoins, être manipulés par le mécanisme de changement de langue de `babel`. La commande peut être utilisée explicitement, dans ce but, mais assurez-vous que cela n'est fait qu'une seule fois pour chaque extension de fichier.

```
\tocbasic@starttoc{prolongation}
```

Cette déclaration est en fait le remplacement de la commande `\@starttoc` à partir du noyau de  $\LaTeX$ . C'est la commande après `\listoftoc*` (voir article 15.2). Les auteurs de classes ou de paqs qui veulent profiter de `tocbasic` vont exploiter cette déclaration, mais il vaut mieux utiliser `\listoftoc`.

La commande `\tocbasic@starttoc` utilise `\starttoc` en interne, mais définit `\parskip` et `\parindent`, avant, à 0 et `\parfillskip` de 0 jusqu'à l'infini.

En outre, `\@currentx` est réglé sur l'extension du fichier en cours disponible entre crochets pour être exécuté par la suite, avant et après la lecture des fichiers d'aide.

$\LaTeX$  ouvre immédiatement un nouveau fichier d'aide pour l'écriture après avoir lu ce fichier, l'utilisation de `\tocbasic@starttoc` entraîne un message d'erreur tel que :

```
Pas de place pour une nouvelle écriture
\ch@ck.. \else \errmessage{Pas de place pour un nouveau 3 #}
\fi
```

exprimé lorsque aucun fichier d'écriture n'est disponible.

La solution peut être trouvée dans ce cas en utilisant le paq `scrwfile` décrit au chapitre 14.



```
\tocbasic@@before@hook
\tocbasic@@after@hook
```

Le crochet `\tocbasic@@before@hook` est exécuté immédiatement avant de lire le fichier d'aide d'une table des matières ou liste de répertoire et avant même l'exécution des instructions d'une commande définies par `\BeforeStartingTOC`.

Il est possible d'étendre ce crochet en utilisant `\g@addto@macro`. KOMA-Script utilise ces crochets pour permettre une adaptation dynamique et fournir le calcul automatique de largeur de la place nécessaire par rubrique de numéros.

Ils sont utilisés seulement pour les classes et, sous réserve, pour les paqs. Les utilisateurs se limiteront à `\BeforeStartingTOC` et `\AfterStartingTOC`. Les auteurs de paqs doivent également utiliser de préférence ces deux instructions.

Les éditions à l'intérieur des deux crochets ne sont pas autorisées !

Si aucune des commandes `\listofeachtoc`, `\listoftoc` et `\listoftoc*` n'est utilisée pour la sortie d'une table des matières ou une autre liste, les crochets doivent être exécutés de manière explicite.

```
\tb@Extension de fichier@before@hook
\tb@Extension de fichier@after@hook
```

Ces instructions sont exécutées sitôt après `\tocbasic@@before@hook` ou avant `\tocbasic@@after@hook`, respectivement pour chaque répertoire qui contient l'extension de fichier. De même `\tocbasic@@after@hook` est exécutée après la lecture d'un tel fichier d'aide et avant les instructions de `\AfterStartingTOC`<sup>1</sup>. Les auteurs de classes et paqs ne doivent jamais les modifier !

Mais si, ni `\listofeachtoc`, ni `\listoftoc`, ni `\listoftoc*` n'est utilisé pour l'édition d'une table des matières ou d'une liste quelconque, les crochets, s'ils sont définis, doivent être exécutés de manière explicite et les instructions peuvent être indéfinies.

Pour un test correspondant voir `\scr@ifundefinedorrelax`, chapitre 12.3.

```
\tocbasic@listhead{titre}
```

La commande, utilisée par `\listoftoc` et `\listofeachtoc` permet l'insert de l'instruction de définition du titre d'un répertoire ou d'une liste. Cela peut être la déclaration du paq `tocbasic` ou le titre d'une rubrique définie séparément, ou encore un titre par défaut. Si vous définissez vos propres instructions, vous pouvez également utiliser `\tocbasic@listhead`. Dans ce cas, avant d'utiliser

1. PAPIRAY précise : la commande est `tocbasic` en toutes lettres dans la version anglaise et serait `tb` dans la version allemande.

`\tocbasic@listhead`, vous devez définir `\@currentx` comme extension de fichier du fichier d'aide correspondant.

```
\tocbasic@listhead@extension de fichier{titre}
```

Cette instruction personnalisée est utilisée dans `\tocbasic@listhead` pour déterminer l'entête de répertoire, les rubriques individuelles ou les entrées `toc` optionnelles sinon elle sera définie, appelée automatiquement et utilisée dans `\tocbasic@listhead`.

```
\tocbasic@addxcontentsline{extension fichier}{niveau}{numéro}{entrée}  
\nonumberline
```

La commande `\tocbasic@addxcontentsline` crée une entrée numérotée de texte au fichier `toc` avec l'extension donnée dans le `niveau` spécifié. L'entrée est numérotée si le numéro hiérarchique de l'argument n'est pas vide et vous pouvez utiliser un argument vide si vous ne voulez pas de numéro. Dans le cas d'un argument vide l'entrée d'un `\nonumberline` n'est préfixée d'aucun argument. Sinon `\numberline` est utilisé avec le numéro de classification comme argument. La déclaration `nonumberline` dans `\listoftoc` est redéfinie en fonction de la ligne de numéro de fonction (voir chapitre 15.2). Ainsi, l'attribut `fixer` ou `effacer` l'ensemble affecte déjà la prochaine exécution de  $\text{\LaTeX}$ .

Noter que la fonction `\tocbasic@addxcontentsline` est obligatoire.

## 14.5 Un exemple complet

Ce chapitre présente l'exemple complet d'un environnement comportant une liste de type flottant définie par l'utilisateur avec `KOMA-Script` en utilisant `tocbasic`. Les commandes internes ont un « @ » dans leur nom, cela signifie que les instructions codées doivent être employées dans un paq séparé, placées entre `\makeatletter` et `\makeatother`, pour développer ou créer sa propre classe. Un nouvel environnement flottant est nécessaire pour fournir figures ou tables. Cela pourrait tout simplement être fait avec :

```
\newenvironment{remarkbox}{%  
\@float{remarkbox}%  
}{%  
\end@float  
}
```

Le nouvel environnement s'appelle `remarkbox`. Chaque figure, table, flottant a un placement par défaut. construit par certaines options de placement bien connues :

```
\newcommand*\fps@remarkbox}{tbp}
```

Ainsi, le nouvel environnement flottant doit être placé par défaut soit en haut d'une page, soit au bas d'une page ou sur une page séparée. Les environnements flottants ont un type flottant numérique. Mais, avec le même bit actif de type flottant, ils ne peuvent pas changer leur ordre.

Figures et tables utilisent normalement les types 1 et 2. Ainsi, un chiffre qui vient plus tard dans le code source d'une table, peut être édité plus tôt que la table et vice versa.

```
\newcommand*\ftype @remarkbox}{4}
```

Le nouvel environnement flottant est de type 4, de sorte qu'il peut dépasser table et figure et peut être dépassé par celles-ci.

Les environnements flottants ont aussi une légende numérotée.

```
\newcounter{remarkbox}
\newcommand*\remarkboxformat}{%
remarque~\theremarkbox\csname autodot\endcsname}
\newcommand*\fnum@remarkbox}{\remarkboxformat}
```

Tout d'abord un nouveau compteur est déterminé, indépendant des chapitres ou autres compteurs de la structure. Il définit  $\LaTeX$  égal aussi à `\theremarkbox` avec une édition standard et des chiffres arabes qui seront ensuite utilisés dans la description de l'édition formatée.

La sortie formatée est à son tour définie comme environnement variable numéroté pour une utilisation dans `\caption`. Les environnements flottants sont identifiés par un nom de répertoire `\jobname` et une extension de fichier.

```
\newcommand*\ext@remarkbox}{lor}
```

L'extension de fichier d'aide pour la liste `remarkbox` est « `lor` », définition de l'environnement flottant. La liste des légendes de cet environnement manque mais `tocbasic` est utilisé pour faciliter sa mise en œuvre qui est documentée dans `\usepackage{tocbasic}` dont le préambule sera utilisé par un auteur de classe ou de paq.

```
\RequirePackage{tocbasic}
```

à la place permet d'enregistrer l'extension de nom de fichier et un titre de répertoire pour le paq `tocbasic`.

`\addtotoclist[float]{lor}`

utilisé en tant que propriétaire de `float`, établit le lien qui appelle les options de KOMA-Script qui se réfèrent à des répertoires de flottants, même dans le nouveau répertoire.

Il permet de définir un titre pour ce répertoire ou un intitulé pour la liste des `remarkbox` :

`\newcommand*{\listoflorname}{liste des jeux d'instructions}`

Le paq `scrbase` est à utiliser pour définir des titres (par défaut en anglais) dans toutes les autres langues que vous souhaitez soutenir. Une commande est nécessaire pour positionner les entrées dans la liste des jeux d'instructions. Voir chapitre 12.4.

`\newcommand*{\l@remarkbox}{\l@figure}`

Voici simplement les entrées dans la liste des `remarkbox` pour obtenir la même disposition que les entrées dans la table des figures. C'est la solution la plus simple. Une plus explicite serait, par exemple :

`\newcommand*{\l@remarkbox}{\@dottedtocline{1}{1em}{1.5em}}`

Vous pouvez utiliser les entrées de chapitre qui ont un impact sur le répertoire. En outre, vous pouvez structurer la liste `remarkbox` selon les chapitres :

`\setuptoc{lor}{chapteratlist}`

La définition de cette propriété le permet lorsque vous utilisez KOMA-Script et toute autre catégorie qui prend en charge cette fonctionnalité. Les classes `standard` ne sont pas incluses et ne l'autorisent pas, mais cela reste suffisant. L'utilisateur dispose, à l'aide des options d'une classe de KOMA-Script ou de `\setuptoc`, de diverses formes de positions (avec ou sans table d'entrée, de contenu, de répertoire, de table des matières, d'entrée TOC numérotée), les sélectionner et éditer le répertoire `\listoftoclistof!\listoftoc{lor}`.

Mais une simple `\newcommand*{\listofremarkboxes}{\listoftoc}{lor}` peut en rendre son utilisation encore plus facile.

Comme vous l'avez vu, cinq commandes se réfèrent à la liste des remarques et seules trois d'entre elles sont nécessaires.

La nouvelle liste prévoit une numérotation facultative de la rubrique avec une entrée non numérotée en option dans la table des matières. Un niveau de la structure inférieure du document, également en option, peut être utilisé pour la rubrique. Les titres d'entête sont fournis avec les classes de KOMA-Script `standard` et avec toutes les classes qui soutiennent explicitement `tocbasic`. Le

support de classes se réfère au nouveau répertoire, même lors du passage à un nouveau chapitre. Les changements de la langue courante sont traités dans la liste des remarques comme ils le sont à l'intérieur de la liste des figures ou de la liste des tables.

En outre, l'auteur d'un paq peut ajouter plus de fonctionnalités.

Ce peut être, par exemple, des options pour cacher l'utilisation de `\setuptoc` à l'utilisateur. Il est intéressant de noter que l'emploi de `tocbasic` peut être référencé pour décrire les fonctions correspondantes. L'avantage de cette situation est, pour l'utilisateur de pouvoir obtenir des informations sur les nouvelles fonctions fournies par `tocbasic`. Il doit, en outre, être en mesure de définir les caractéristiques des remarques, même sans connaissances sur une simple extension de fichier lor :

```
\newcommand*{\setupremarkboxes}{\setuptoc{lor}}
```

passé les arguments des propriétés directement à la liste `\box!setupremarkboxes` comme une liste des caractéristiques de l'extension de fichier lor.

## 14.6 Tout dans une seule instruction

L'exemple de la section précédente montre qu'il est assez facile avec `tocbasic` de définir vos propres flottants avec leurs propres répertoires. L'exemple suivant révèle que ce peut être encore plus facile.

```
\DeclareNewTOC[liste des options]{extension de fichier}
```

Cette commande d'extension déclare en une seule étape une nouvelle liste, le titre de cette liste, le terme utilisé pour les entrées dans la liste, et gère l'extension de nom de fichier.

En outre les options flottant et non-flottant peuvent être définies, et `\caption` peut être utilisé à l'intérieur de ces environnements. Les fonctions ajoutées `\captionabove`, `\captionbelow` et `\captionbeside` des classes de KOMA-Script (voir section 3.20) peuvent également être utilisées dans ces environnements.

Comme déjà expliqué dans la section 14.1, le nom de fichier du répertoire en cours de création est « extension de fichier ». Cet argument est obligatoire et ne doit pas être vide. La liste d'options de l'argument est une liste séparée par des virgules, comme dans toutes les listes de `\KOMAOPTIONS` (voir section 2.4), mais ces options ne peuvent pas être réglées à l'aide de `\KOMAOPTIONS`.

Un aperçu de toutes les options disponibles peut être trouvé dans la table 14.3 page 457.

L'exemple de l'article 14.4 peut être considérablement réduit en utilisant la nouvelle déclaration `\DeclareNewTOC`.

```
\DeclareNewTOC[%
    type=remarkbox,%
    types=remarkboxes, %
    float, % l'environnement flottants est à définir.
    floatype=4,%
    name=repère,%
    listname={liste des repères \"théorème}%
]{lor}
\setuptoc{lor}{chapteratlist}
```

Le compteur `remarkbox` inclut les commandes `\remarkbox` et `\remarkbox*`, ainsi que `\theremarkbox`, `\remarkboxname` et `\remarkboxformat` utilisées pour des sous-titres, certaines commandes internes faisant référence à l'extension de fichier `lor` et les commandes `\listofremarkboxes` utilisée à la liste de remarques et `\listremarkboxnames` utilisée côté environnements pour une meilleure compréhension de leurs usages possibles définis dans la table 14.2 page ci-contre avec une comparaison des instructions et environnements créés par exemple avec la commande `remarkbox` dans un nouvel environnement pour figures.

TABLE 14.3 – Options pour la commande `\DeclareNewTOC`

Instructions et environnements
<code>atbegin=Code</code> Si un nouvel environnement flottant ou non flottant est défini, de sorte que chaque <code>code</code> est exécuté à la fin de cet environnement.
<code>atend=Code</code> Si un nouvel environnement flottant ou non flottant est défini, de sorte que chaque <code>code</code> est exécuté à la fin de cet environnement.

---

 Instructions et environnements (suite)
 

---

**counterwithin=compteur**  $\LaTeX$

Si vous définissez une nouvelle figure ou un table flottant (ou non-flottant), les légendes seront numérotées et un type de compteur sera spécifié. Vous pouvez déclarer un autre compteur pour le compteur parent  $\LaTeX$ . Ce compteur sera utilisé de la même manière que, par exemple, le compteur des figures de la classe `book` dont dépend la numérotation des chapitres faite avec  $\LaTeX$ .

**float**

Si `float.` est défini, l'environnement flottant sera aussi défini. Il ne définit pas seulement un nouveau type de répertoire, mais également le type de flottants (voir type d'option) et le type d'entrée\* (voir, figure et figure\*).

**floatpos=position du flottant**

Chaque figure ou table a un comportement de positionnement prédéfini qui peut être modifié par l'utilisation d'un argument optionnel du flottant. Avec cette option, le comportement de glissement est fixé pour les figures et les tables en création (voir l'option `float`). La syntaxe et la sémantique sont les mêmes que celles de l'argument facultatif pour les figures et les tables. Si l'option n'est pas utilisée, la valeur `tbp` est appliquée par défaut, comme pour les classes `standard` des figures et tables : `top`, `bottom` et `page`. (cette valeur pourrait être francisée en `hbp` [haut - bas - page]).

**floattype=nombre**

Chaque figure ou table a un type numérique. Les flottants dans lesquels seuls les bits différent sont liés à ce type de glissement et peuvent dépasser les uns des autres. Les types de flottants avec bits communs ne peuvent pas être réorganisés. Le flottant figure est généralement de `type1` et le flottant table de `type2`. Des types autorisés existent de 1 à 31 (tous les bits sont utilisés, de sorte qu'ils puissent dépasser et être dépassés par n'importe quel autre type). Si aucun type n'est spécifié, le type le plus élevé - 16 - d'un bit est utilisé.

**forcenames**

Voir options `name` et `listname`.

**hang=retrait**

---

---

**Instructions et environnements (suite)**

---

Cette option est obsolète depuis KOMA-Script 3.20. La largeur du nombre de l'entrée de répertoire est maintenant considérée comme une propriété la remplaçant, selon le style de la liste spécifié de répertoire de l'option `tocentrystyle`. Avec les styles de KOMA-Script, la propriété serait, par exemple, `numwidth` et avec, l'option `tocnumwidth`. Si un style propriété est pré-réglé par `\DeclareNewTOC` avec `1,5em`, il peut être facilement modifié en spécifiant explicitement `tocnumwidth=valeur` avec une valeur différente. Pour les figures, utiliser par exemple la valeur de `2.3em`.

---

**`indent=retrait`**

---

Cette option est obsolète depuis KOMA-Script 3.20. La valeur numérique de l'entrée gauche du répertoire est considérée comme propriété la remplaçant, selon le style spécifié de la liste de répertoire de l'option `tocentrystyle`. Avec les styles de KOMA-Script, la propriété serait par exemple `indent` et avec, l'option `tocindent`. Si un style de cette propriété est pré-réglé par `\DeclareNewTOC` avec `1em`, il peut être modifié en spécifiant explicitement `tocindent=valeur` avec une valeur différente. Pour les figures, utiliser par exemple la valeur de `1.5em`.

**`level=niveau hiérarchique`**

Dans cette option obsolète depuis KOMA-Script 3.20, la valeur numérique du niveau de l'entrée de répertoire est considérée comme une propriété, selon le style spécifié de la liste de répertoire de l'option `tocentrystyle`. Néanmoins, tous les styles ont le niveau de la propriété et donc l'option `toclevel`. La propriété est pré-réglée par `\DeclareNewTOC` à `1`. Ce défaut peut être remplacé par une autre valeur en spécifiant explicitement `toclevel=valeur`

---

**`listname=chaîne (titre de répertoire)`**

---

Chaque répertoire titre a un titre qui peut être déterminé par cette option. Si l'option n'est pas spécifiée, nous utiliserons un répertoire intitulé « Liste de divers types d'entrée » (voir l'option `types`), où le premier caractère de la dénomination du type d'entrée est converti en lettres majuscules. Il existe aussi une macro `\listedesnomsdetrypesdentrées` définie avec cette valeur qui peut être modifiée à tout moment. Cette macro n'est définie que si elle ne l'est pas déjà ou que l'option supplémentaire `forcenames` est encodée. Si aucun nom n'est donné, la valeur de type avec le premier caractère en majuscule sera utilisée.

---

**`name=chaîne (nom de l'entrée)`**

---

Nom utilisé à la fois comme préfixe facultatif pour les entrées du répertoire et pour l'étiquette dans un environnement flottant (voir l'option `float`)

---



---

**Instructions et environnements (suite)**

---

ou non flottant (voir l'option `nonfloat`). Sans cette option, le nom d'entrée (voir l'option `type` d'entrée) est utilisé, avec le premier caractère converti en majuscules. Il existe aussi une macro `\nomdutypedentrée` définie avec cette valeur qui peut être modifiée à tout moment. Cette macro n'est définie que si elle ne l'est pas déjà ou que l'option supplémentaire `forcenames` est encodée.

---

**`nonfloat`**

---

S'il est défini, il détermine, non seulement un nouveau type de répertoire, mais aussi un environnement non flottant de `listing` (voir `Optiontype`) similaire à un flottant qui peut être utilisé dans les limites de l'environnement en cours de validité.

---

**`owner=chaîne` (propriétaire)**

---

Chaque nouveau répertoire a un propriétaire qui peut être indiqué ici, avec `tocbasic` (voir la section 14.1). Si aucun propriétaire n'est spécifié, c'est le propriétaire « float » qui est utilisé avec les classes de KOMA-Script pour les répertoires des figures et des tables.

---

**`tocentrystyle=style d'entrée`**

---

Le style que les entrées devraient avoir dans le répertoire approprié. Le nom du niveau d'entrée est déterminé par l'option `type`. En plus des options de la table, toutes les propriétés du style peuvent être spécifiées comme options par les noms des propriétés avec le préfixe `toc`. Ainsi, la valeur numérique du niveau peut être spécifiée par exemple sous forme `toclevel`. Pour plus d'informations sur les styles, voir le chapitre « configuration des entrées ».

---

**`tocentryStiloption=valeur`**

---

En fonction de la sélection par `tocentrystyle` (voir la section 15.3), d'autres options de style d'entrée. La liste `tocbasic` des styles d'entrée de répertoire prédéfinis utilisables comme options d'attributs de style se trouve dans la table 14.1 page 442.

---

**`type=type d'entrée`**

---

Spécifie le type d'entrée dans le répertoire approprié, également utilisé comme nom de base pour différentes macros, environnements et compteurs et ne doit donc contenir que des lettres. Si l'option est utilisée, l'extension de fichier de l'argument obligatoire le sera pour le type d'entrée.

---

**`types=type d'entrée multiples`**

---

À divers moments, la forme plurielle du type d'enregistrement est utilisée pour définir, par exemple, une commande `\listof` de type d'entrée multiple. Si l'option n'a pas été utilisée, elle sert pour un type d'entrée simple.

---

TABLE 14.2 – Comparaison des environnements échantillons et figures

environnement	figure	NewToc	description
<code>remarkbox</code>	<code>figure</code>	type, float	environnement particulier de flottant
<code>remarkbox*</code>	<code>figure*</code>	type, float	créer des colonnes sur chaque type de flottant
<code>remarkbox</code>	<code>figure</code>	type, float	compteur utilisé par <code>\caption</code>
<code>\theremarkbox</code>	<code>\thefigure</code>	type, float	déclaration à la sortie du compteur
<code>\remarkboxformat</code>	<code>\figureformat</code>	type, float	instructions de formatage de chaque compteur à la sortie de <code>\caption</code>
<code>\remarkboxname</code>	<code>\figurename</code>	type, float, name	nom de l'étiquette utilisée dans <code>\caption</code>
<code>\listofremarkboxes</code>	<code>\listoffigures</code>	type, float	déclaration à la sortie de chaque répertoire
<code>\listremarkboxnames</code>	<code>\listff@figurename</code>	type, float, listname	en route pour le répertoire
<code>\fps@remarkbox</code>	<code>\fps@figure</code>	type, float, floattype	numérotation séquentielle de flottants
<code>lor</code>	<code>lof</code>		extension fichier auxiliaire pour répertoire

Le code utilisé pour l'environnement suivant ressemble à ceci :

```
\begin{remarkbox}[!]  
\centering  
\fbox{\parbox{.85\textwidth}  
{Le même élément devrait toujours être typographié de la même  
manière avec une même mise en forme et la même apparence.}  
}  
\caption{Première règle de typographie}  
\label{remA}  
\end{remarkbox}
```

Il permet d'obtenir :

Le même élément devrait toujours être typographié de la même manière avec une même mise en forme et la même apparence
---

mnémotechnique 1 – Première règle de typographie



# Chapitre 15

## Améliorer les « paqs » étrangers avec `scrhack`

Certains paqs ne fonctionnent pas bien avec `KOMA-Script` et pourraient être améliorés, mais il est souvent très difficile d'en convaincre leurs auteurs. Cela vaut également pour les paqs dont le développement a été arrêté. C'est pour cette raison que le paq `scrhack` a été développé.

Il permet de modifier les instructions et les définitions d'autres paqs, afin qu'ils puissent mieux travailler avec `KOMA-Script`.

Les redéfinitions ne sont activées que si ces paqs ont été chargés. L'utilisateur peut empêcher `scrhack` de redéfinir les macros de paqs individuels.

### 15.1 Note sur le développement

Bien que ce paq soit intégré depuis plusieurs années à `KOMA-Script` et qu'il soit adopté par de nombreux utilisateurs, il présente aussi un problème : dans la redéfinition des macros, il dépend non seulement de leur utilisation exacte, mais aussi de la version spécifique du paq de ces macros. Donc, `scrhack` doit être adapté, en permanence, aux nouvelles versions des autres paqs, encore et encore, et ne peut jamais être considéré comme terminé.

Par conséquent, la version bêta de *scrhack* est une version durable. Bien que son utilisation apporte généralement certains avantages, la fonction ne peut pas être garantie en permanence.

Si une version inconnue des paqs correspondants est utilisée et ne peut pas s'exécuter, *scrhack* applique une politique de patch nécessaire. Dans un cas extrême, un patch avec une version inconnue peut au contraire entraîner une erreur.

## 15.2 Options de sélection précoce ou tardive

Tout ce qui est décrit dans la [section 2.4](#) est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la [section 2.4](#), vous pouvez continuer à la [section 16.3](#).

Cette partie décrit une caractéristique du programme présenté et concerne non seulement *scrhack* mais aussi d'autres paqs et classes *KOMA-Script*. Ainsi, l'utilisateur trouvera toutes les informations sur un paq ou une classe dans cette [section](#) qui figure presque à l'identique dans plusieurs chapitres. Les utilisateurs seulement intéressés par le guide sur un paq ou une classe et par une vue d'ensemble de *KOMA-Script* peuvent la lire puis l'ignorer dans une étude plus approfondie du manuel.

```
\documentclass[liste des options]{KOMA-Script-classe}
\usepackage[liste des options]{liste des paqs}
```

Dans  $\text{\LaTeX}$ , les utilisateurs spécifient les options de classe sous la forme d'une liste de mots clés simples séparés par une virgule comme argument optionnel de `documentclass`. Sauf que ces options sont également transmises à toutes les classes et à tous les paqs qui comprennent ces options. De même, il est admis que les utilisateurs spécifient comme argument optionnel de `usepackage` les options de paqs sous la forme d'une simple liste de mots-clés séparés par des virgules.

*KOMA-Script* étend ces possibilités en intégrant une valeur supplémentaire qui peut prendre la forme de `option=valeur`, utilisable indistinctement avec `documentclass` et `usepackage`*KOMA-Script* ou avec  $\text{\LaTeX}$  comme décrit par exemple dans [[DGS+12](#)] et [[Tea05b](#)].

Le réglage des options a cependant, avec `documentclass` et `usepackage`, un inconvénient qui doit être observé : commandes, longueurs, compteurs... peuvent facilement casser. Ainsi, l'utilisation d'une longueur  $\text{\LaTeX}$  comme valeur d'une

option entraîne dans ces instructions un message d'erreur pour reprendre le contrôle avant que la valeur de `KOMA-Script` ne soit adoptée. L'affectation de valeurs pouvant être réalisées avec des longueurs de `LATEX` ou des instructions `KOMAoptions` et `KOMAoption` est documentée ci-dessous.

```
\KOMAoptions{liste d'options}  
\KOMAoption{option}{liste des valeurs}
```

`KOMA-Script` offre, avec l'instruction `KOMAoptions` la possibilité de changer la valeur de la plupart des options de classes et de paqs, même après le chargement de la classe ou du paq. Vous pouvez rectifier les valeurs qui seront récupérées par chacune des options appartenant à la liste pour un certain nombre d'options disponibles. Certaines ont aussi une valeur par défaut (eng. *Default Value*). Si on omet de renseigner une valeur, c'est la valeur par défaut qui est appliquée automatiquement.

Dans certains cas, elles peuvent contenir simultanément plusieurs valeurs, avec la possibilité d'assigner à `KOMA-Script` une option séquentielle d'un ensemble de valeurs. Les valeurs individuelles sont, dans la liste de valeurs, séparées les unes des autres par des virgules.

Si une valeur contient un signe égal ou une virgule, alors la valeur est mise entre accolades.

`KOMA-Script` utilise pour la réalisation de cette possibilité les commandes `FamilyOptions` et `FamilyOption`, voir dans la partie II, la section 11.3.

## 15.3 Utilisation de tocbasic

Dans les premiers temps de `KOMA-Script`, les utilisateurs ont demandé que les répertoires de flottants générés en utilisant le paq `float` soient traités comme les flottants, telles la liste des figures et la liste des tables, qui sont générées par `KOMA-Script` lui-même.

À cette époque, les auteurs de `KOMA-Script` et de `float` ont collaboré pour concevoir une interface appropriée à cette extension, finalement réalisée sous la forme des deux commandes `\float@listhead` et `\float@addtolists`.

Mais elles ne sont pas suffisamment flexibles pour supporter l'ensemble des fonctionnalités offertes par `KOMA-Script` et malheureusement, l'auteur de `float` en avait déjà finalisé le développement ce qui rendait le paq peu susceptible de déboucher sur d'autres modifications.

D'autant plus qu'entre-temps, d'autres auteurs de paqs avaient adopté et adapté les deux déclarations. Il est aussi apparu que la mise en œuvre dans certains paqs, `float` compris, nécessite un certain ordre de chargement des paqs, même s'il ne sont pas liés les uns aux autres. Une erreur peut être provoquée par un ordre de chargement incorrect qui peut même casser la fonctionnalité des commandes.

Officiellement, `KOMA-Script` ne supporte plus cette ancienne interface pour effacer tous ces inconvénients et problèmes, mais avertit si l'ancienne interface est utilisée.

Dans le même temps, le paq `tocbasic` (voir chapitre 14) a été conçu et mis en œuvre comme une interface centrale pour la gestion de la table des matières, des listes de flottants et similaires. Son utilisation fournit beaucoup plus d'avantages et de fonctionnalités que les deux commandes ci-dessus.

Bien que l'effort requis pour adopter ce paq soit très faible, les auteurs de la plupart des paqs, qui utilisent l'ancienne interface, ne l'ont pas fait pour le moment. Par conséquent `scrhack` contient lui-même des ajustements appropriés pour les paqs `float`, `floatrow` et `listings`.

Le chargement `scrhack` est suffisant pour que ces paqs reconnaissent non seulement les réglages de `KOMA-Script`, l'option `listOf`, mais aussi le change de langue du paq `babel`.

Pour plus d'informations sur les options qui sont maintenant disponibles sur `tocbasic`, consulter la section 14,2.

Si la modification de l'un des paqs n'est pas souhaitée ou provoque des problèmes, elle peut être désactivée sélectivement avec l'option `float=false`, `floatrow=false`, ou `listings=false`.

Il est important de noter que changer des options après le chargement du paq associé n'a aucune influence.

## 15.4 Faux espoirs avec `\@ptsize`

Certains paqs supposent fondamentalement que la macro interne de la classe `\@ptsize` est à la fois fixée et étendue à un nombre entier. Pour des raisons de compatibilité `KOMA-Script` définit `\@ptsize` avec d'autres tailles de polices de base en 10 pt, 11 pt, ou 12 pt et permet également des tailles de police fractionnaires `\@ptsize` étendues naturellement à une fraction décimale.



L'un des paqs qui ne peut pas l'inclure, est le paq `setspace` dont l'ensemble des valeurs est toujours dépendant de la taille de base de la police, même si le réglage est effectué dans le cadre d'une taille de police différente. Le paq `scrhack` résout les deux problèmes à l'aide des paramètres de `\onehalfspacing` et `\doublespacing` en procédant toujours par rapport à la taille courante réelle de la police. Si ce changement n'est pas souhaité, ou cause des problèmes, il est nécessaire de paramétrer sélectivement `setspace=false`. Le changement de l'option après le chargement de `setspace` ne doit avoir aucune influence et pour cette raison, `scrhack` doit être chargé avant `setspace` afin que l'une des options de `onehalfspacing` ou `doublespacing`, n'affecte pas le chargement.

## 15.5 Cas particulier avec hyperref

Avant la version 6.79h, le paq `hyperref` définissait les liens d'ancrage après plutôt qu'avant la rubrique des commandes de versions étoilées comme `\chapter*`, `\part*`, etc... Ce problème avait été résolu par une suggestion de l'auteur de `KOMA-Script`, en ajoutant à `scrhack` un patch correspondant qui peut être désactivé par `hyperref=false`.

Il est néanmoins recommandé d'utiliser la version actuelle de `hyperref`. Dans ce cas `scrhack` désactive la modification qui n'est plus nécessaire.

## 15.6 Traitements incompatibles de `\textwidth` et `\textheight`

Le paq de `landscape` définit l'environnement au format paysage, mais ne gère pas l'entête, ni le pied. Dans cet environnement `\textheight` est réglé sur la valeur de `\textwidth`, mais `\textwidth` n'est pas réglé sur la valeur précédente de `\textheight`. Cela est incompatible. À ma connaissance, `\textwidth` n'est pas modifié en conséquence, car d'autres paqs ou des instructions d'utilisation pourraient en être perturbés.

Un changement de `\textwidth` peut endommager la fonction, par exemple, des paqs `showframe` et `sclayer`.

Par conséquent, il est préférable que `\textheight` reste inchangé, que `scrhack` utilise le paq `xpatch` (voir [Gre12]) pour la commande de démarrage dans `\landscape` et pour changer le milieu en utilisant le même nom.

Si ce changement n'est pas souhaité ou est à l'origine des problèmes, il peut être désactivé avec l'option `lscap=false`.

Il est bon de noter qu'une affectation ultérieure de l'option `lscap` avec `\KOMAOption` ou `\KOMAOptions` n'a d'effet que si `lscap` n'était pas, pendant son chargement, `false`. Le paq `lscap` est également utilisé par `pdfscape` de sorte que *scrhack* influence également la fonction de ce paq.

## 15.7 Cas spécial de nomencl

Une caractéristique spéciale décrit le `hack` pour le paq `nomenc1`. D'une part, une mise à niveau après observation de la table en option d'entrée de contenu pour l'option de nomenclature `toc=indentunnumbered` quasi incidemment mais aussi égales aux terminaisons `nlo` et `nls` réservées dans le paq `tocbasic` (voir `\addtotoclist`, section 14.1). En outre, divers attributs pour l'extension `nls` sur `\setuptoc` peuvent être définis à l'aide du `hack`. Ainsi, il est par exemple possible non seulement d'entrer dans la nomenclature `\setuptoc{nls}{numéro}` de la table des matières, mais aussi de porter le même numéro. Un autre effet de la nomenclature avec ce `hack` permet de recevoir également un titre de colonne approprié si la rubrique `kolumntitel` est activée par exemple en utilisant le style de page `headings`. Tout ceci est réalisé avec un petit patch de l'environnement `thenomenclature`.

# Chapitre 16

## Définition des niveaux et styles de page avec `scrlayer`

Les utilisateurs de programmes graphiques sont déjà familiers avec le modèle de niveaux pour une page, qui reste assez étrange en ce qui concerne `LATEX`. Mais si `LATEX`, lui-même, ne connaît pas les couches, il existe des paqs comme `esopic` ou `textpos`, qui lui apportent un fond ou une couche de premier plan.

Dans `KOMA-Script` `scrlayer` existe un paq qui fournit également cette toile de fond ainsi que les couches de premier plan, avec une différence majeure vis à vis des autres paqs mentionnés ci-dessus, elles font partie de la définition du style de page, ce qui permet, tout simplement, de passer d'une couche à l'autre.

Pour ce faire, l'ensemble prend également en charge une interface de bas niveau pour définir des styles de page en utilisant un empilement de couches, dans des mots courts, et gérer cet empilement de couches :

- mettre des couches sur la pile de couches,
- mettre des couches à la position la plus basse de la pile de couches,
- mettre des couches avant ou après une couche de la pile de couches,
- enlever une couche de la pile de couches,
- supprimer des doublons de couches de la pile de couche.

Néanmoins, il est recommandé aux utilisateurs avancés d'utiliser directement les couches.

Des interfaces pour débutants ou utilisateurs moyens sont fournies par des paqs supplémentaires, qui, à leur tour, chargent `sclayer`. Voir le chapitre 6 de la partie I de ce manuel.

## 16.1 Note sur le niveau de développement

Ce paq est en cours de développement. Certaines parties en sont même au stade expérimental, en particulier les fonctions internes peuvent être modifiées à l'avenir et le paq sera très probablement étendu. Et à cause de cet état inachevé de développement, ne vous attendez pas à un manuel d'utilisation complet et fini, mais néanmoins, ce manuel pour les utilisateurs et les développeurs avancés décrit l'état actuel de développement et les parties libérées recommandées de `sclayer`.

Il est fortement déconseillé d'utiliser les éléments qui ne sont pas documentés ici à autre chose qu'à des tests.

## 16.2 Choix de l'option de sélection précoce ou tardive

Ce qui est écrit dans la `section 2.4` est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la `section 2.4`, vous pouvez continuer `section 17.3`.

Cette `section` décrit une caractéristique du programme présenté concernant non seulement `sclayer`, mais aussi d'autres paqs et classes KOMA-Script. Ainsi, l'utilisateur trouvera toutes les informations sur un paq ou une classe dans ce paragraphe qui figure presque à l'identique dans plusieurs chapitres. Les utilisateurs non intéressés par une étude approfondie de l'orientation d'un paq ou d'une classe mais désirent seulement obtenir une vue d'ensemble de KOMA-Script peuvent lire cette `section` dans un des chapitres, puis l'ignorer.

```
documentclass[liste des options]{KOMA-Script-classe}
usepackage[liste des options]{liste des paqs}
```

Dans L<sup>A</sup>T<sub>E</sub>X, les utilisateurs spécifient les options de classe sous la forme d'une liste de mots clés simples séparés par une virgule comme argument optionnel de `documentclass`. Sauf que ces options sont également transmises à toutes les

classes et à tous les paqs qui comprennent ces options. De même, il est admis que les utilisateurs spécifient comme argument optionnel de `usepackage` les options de paqs sous la forme d'une simple liste de mots-clés séparés par des virgules.

KOMA-Script étend ces possibilités en intégrant une valeur supplémentaire qui peut prendre la forme de `option=valeur`, utilisable indistinctement avec `documentclass` et `usepackage`KOMA-Script ou avec L<sup>A</sup>T<sub>E</sub>X comme décrit par exemple dans [DGS+12] et [Tea05b].

Le réglage des options a cependant, avec `documentclass` et `usepackage`, un inconvénient qui doit être observé : commandes, longueurs et compteurs... peuvent facilement casser. L'utilisation d'une longueur L<sup>A</sup>T<sub>E</sub>X comme valeur d'une option entraîne dans ces instructions un message d'erreur pour reprendre le contrôle avant que la valeur de KOMA-Script ne soit adoptée. L'affectation de valeurs qui peuvent être réalisées avec des longueurs L<sup>A</sup>T<sub>E</sub>X ou instructions KOMAoptions et KOMAoption est documentée ci-dessous.

```
KOMAoptions{liste d'options}
KOMAoption{option}{liste des valeurs}
```

KOMA-Script offre, avec l'instruction KOMAoptions la possibilité de changer la valeur de la plupart des options de classes et de paqs, même après le chargement de la classe ou du paq. Vous pouvez rectifier les valeurs qui seront récupérées par chacune des options appartenant à la liste pour un certain nombre d'options disponibles. Certaines ont aussi une valeur par défaut (eng. *Default Value*). Si on omet de renseigner une valeur, c'est la valeur par défaut qui est appliquée automatiquement.

Dans certains cas, elles peuvent contenir simultanément plusieurs valeurs, avec la possibilité d'assigner à KOMA-Script une option séquentielle d'un ensemble de valeurs. Les valeurs individuelles sont, dans la liste de valeurs, séparées les unes des autres par des virgules.

Si une valeur contient un signe égal ou une virgule, alors la valeur est mise entre accolades.

KOMA-Script utilise pour la réalisation de cette possibilité les commandes `FamilyOptions` et `FamilyOption`, voir dans la partie II, la section 11.3.

## 16.3 Informations élémentaires

Le paq nécessite une information générique de base sur la classe utilisée. L'auteur de classes peut aider `sclayer` en l'ajoutant et la rendant pertinente. Sinon le paq essaie lui-même de détecter l'information. Cela fonctionne, par exemple, pour les classes `standard` et les classes de `KOMA-Script`. Mais cela peut échouer partiellement ou complètement avec d'autres classes. Ce chapitre décrit certaines des informations, que les auteurs de classe peuvent fournir.

Généralement les utilisateurs n'ont pas besoin de s'inquiéter à ce sujet.

```
\if@chapter puis le code \else un autre code \fi
```

Si `\if@chapter` est défini et que `\iftrue` correspond, `sclayer` examinera le niveau du chapitre, par exemple en collaboration avec l'option de traitement `automark`.

S'ils sont définis, mais différents de `\iftrue`, `sclayer` traitera seulement les niveaux de commandes `\part`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`, `\subsubparagraph`.

Si la macro n'est pas définie, la question est de savoir si `sclayer` doit traiter le niveau du chapitre, impérativement à la commande `\chapter`.

Si cette déclaration est définie et ne correspond pas à `\relax`, `sclayer` définit la macro `\if@chapter` comme synonyme de `\iftrue`. Sinon `\if@chapter` sera considéré comme un synonyme de `\iffalse`.

```
\if@mainmatter puis le code \else un autre code \fi
```

Les classes telles que `\frontmatter` `\mainmatter` et `\backmatter` de `book` ou `scrbook` sont définies pour basculer entre le début, le corps et la fin d'un livre.

En général, on utilise la classe interne `\if@mainmatter`, afin d'identifier si le sujet en cours est la matière principale travaillée actuellement dans la partie principale du document ou non.

Les classes `scrreprt` (rapport) et `scrartcl` (article) considèrent qu'il n'y a qu'une partie principale, n'utilisent jamais `\frontmatter`, `\mainmatter` ou `\backmatter` et n'exploitent pas `\if@mainmatter`. Elles supposent à la place qu'il n'existe qu'une partie principale.

Pour `sclayer` il est plus facile de ne pas tester systématiquement l'existence des commandes en la matière, mais d'utiliser `\if@mainmatter` même avec des classes comme `scrreprt` et `scrartcl`.

Cela devrait correspondre alors à `\iftrue` dans les catégories ci-dessus, et si `\if@mainmatter` n'est pas précisé, il sera défini comme un synonyme de `\iftrue`.

Cependant, certaines classes prédéfinissent `\frontmatter` `\mainmatter` ou `\backmatter` et pourtant, n'ont toujours pas `\if@mainmatter`.

Dans ce cas `scrlayer` définit `\if@mainmatter` comme synonyme de `\iftrue` et élargit au-delà les définitions de `\frontmatter`, `\mainmatter` et `\backmatter` pour fixer `\if@mainmatter` correctement.

S'il y a d'autres commandes comparables pour la commutation entre partage de documents différents, `scrlayer` ne les connaît pas, ne les teste pas, elles ne sont pas extensibles donc non appropriées, et dans ce cas, `scrlayer` est tributaire de la coopération de l'auteur de classe pour définir `\if@mainmatter` correctement.

`\DeclareSectionNumberDepth{nom de niveau}{profondeur de niveau}`

En général, un nombre entier est attribué à chaque niveau de la `section` pour indiquer sa profondeur dans la structure du document.

`LATEX` en a besoin pour gérer la hiérarchie des niveaux structurels de `section`.

Normalement la classe de document, qui définit les commandes de la `section`, connaît elle-même la profondeur d'un niveau et utilise, en cas de besoin, la valeur numérique à l'intérieur des commandes correspondantes.

Mais `scrlayer` a également besoin des informations sur la hiérarchie de la `section`.

Avec la commande `\DeclareSectionNumberDepth` vous pouvez assimiler le nom d'un niveau de titre à un niveau de profondeur.

Avec la classe `standard book`, le nom des niveaux serait, par exemple, partie, chapitre, `section`, paragraphe, sous-paragraphe ou `alinéa` et leur profondeur respectivement associée de -1, 0, 1, 2, 3, ou 4, `scrlayer` détermine d'abord ces valeurs pendant le chargement du paq, puis les identifie de nouveau lors de `\begin{document}`.

Mais, si elle échoue, c'est-à-dire si des commandes de `section` qui ne conduisent pas à un résultat correct sont utilisées, `\DeclareSectionNumberDepth` donne à l'auteur de la classe la possibilité de définir explicitement le lien entre niveau et numéro.

## 16.4 Déclaration de niveaux

Un niveau (angl. *Layer*), ou couche, est une feuille virtuelle (en face d'une feuille de papier physique) que l'on pourrait aussi définir en tant que modèle conceptuel pour une page.

Contrairement au vrai papier physique, cet écran est transparent dans son ensemble.

Habituellement, plusieurs couches sont empilées, et un matériau opaque sur une couche peut cacher un nombre important de couches inférieures. Cet amas de couches est ensuite *mappé*<sup>1</sup> sur une page afin de créer la page physique.

Le paq `sclayer` fournit deux de ces empilements de couches pour chaque page : un empilement de couches de fond et une pile de couches de premier plan.

La pile de fond est à l'origine du contenu de la page normale, la pile de couches de premier plan est au-dessus du contenu de la page normale.

Ainsi, le contenu normal de la page est une sorte de couche de séparation entre la pile de couches de fond et de la pile de couches de premier plan.

Une couche a plusieurs attributs qui peuvent être compris comme des réponses à des questions fondamentales. Le premier état d'attribut consiste à identifier si la couche fait partie du premier plan ou de l'arrière-plan. Les niveaux de fond sont imprimés en premier, avant le contenu normal de la page. Ils sont suivis par les principaux contenus et les couches de premier plan et apparaissent donc derrière ou en dessous de ce contenu principal.

Par défaut, un plan est composé à la fois d'une couche de fond et d'une couche de premier plan : il est donc imprimé deux fois.

***Dans quel endroit se situe le niveau à émettre ?*** Répondre à cette question définit les positions horizontale et verticale de la couche, réglées par le second attribut pour la position horizontale et par le troisième pour la position verticale.

***Quelle est la taille de la couche ?*** En ce qui concerne la position, l'étendue horizontale et verticale de la couche peuvent être plus petite ou plus grande que le papier et se positionner différemment sur ce même papier.

***Comment les positions horizontale et verticale peuvent-elles être mesurées ?*** Répondre à cette question définit la propriété de l'alignement.

---

1. Anglicisme signifiant « cartographié » et plus simplement « reproduit »



On peut mesurer à partir du bord gauche du papier jusqu'au bord gauche de la couche, à mi-niveau ou encore jusqu'au bord droit de cette couche. En vertical, on peut aussi mesurer depuis le bord supérieur du papier jusqu'au bord supérieur de la couche, jusqu'au niveau moyen ou jusqu'au bord inférieur de cette couche.

*Quel est le niveau de sortie de texte ou du graphique fournis ?* Cette question est étroitement liée à la position. Pour l'utilisateur, il est supposé que, dans l'exemple de sortie graphique, l'origine se situe dans le coin inférieur gauche de la couche, mais que cette disposition serait plutôt défavorable pour la sortie de texte. De là l'origine du calque de texte à hauteur d'une ligne standard depuis le coin supérieur gauche de la couche. Une couche graphique s'allonge d'elle-même dans un environnement image, et des commandes supplémentaires sont disponibles pour le positionnement.

*Le niveau est-il imprimé sur les côtés gauche ou droit d'un document ?* Par défaut, une couche est imprimée sur toutes les pages. Noter que L<sup>A</sup>T<sub>E</sub>X traite les pages, même avec des numéros de page paire sur les pages de gauche et avec des numéros de page impaire sur les pages de droite, mais en mode recto seulement les pages de droite. L<sup>A</sup>T<sub>E</sub>X se réfère à la pratique de la langue anglaise, et traite, en mode recto-verso, les pages avec numéro pair comme page de gauche et celles avec numéro impair comme page de droite, mais qu'il n'y a, en mode recto, ni page de gauche ni page de droite.

*Le niveau doit-il être utilisé dans des documents recto ou étendu au mode recto-verso ?* Par défaut, le niveau est illimitée, de sorte qu'il est délivré à la fois dans le mode unilatéral ou comme mode recto-verso. Néanmoins, un plan limité au côté droit ne sera jamais imprimé en mode unilatéral sans pour autant être considéré comme une couche de mode recto-verso.

*Le niveau est-il consacré à des surfaces avec flottants ou des pages normales ?* L<sup>A</sup>T<sub>E</sub>X produit des surfaces de glissement pour les objets à partir des environnements tels que tableau ou figure si cela a été autorisé, pour gérer, sur des pages séparées, le contenu de documents normaux (voir l'option **p** pour table ou figure). Dans certains cas, il est permis de glisser par un côté dans le document. Sont considérées comme pages normales toutes les pages qui n'ont pas de surface de glissement. Donc, d'un certain point de vue une page de « flottant » peut glisser dans le document. De la même manière, une page « non-flottant » n'est pas une page sans « flottant », mais une page, qui ne glisse pas dans le document. Une page normale peut inclure un « flottant » au début, à l'intérieur ou à la fin. Un très gros « flottant » peut également créer l'impression

d'un coulissant latéral, mais en réalité, il est placé sur le dessus d'une page normale.

**Quel est le contenu de la couche ?** Il s'agit simplement de tout ce qui sera imprimé chaque fois que la couche est éditée.

Nous avons donc neuf questions avec neuf attributs qui se traduisent par neuf propriétés. Dans ce tutoriel, nous décrirons des attributs supplémentaires qui sont définis pour la commodité et peuvent être exprimés par une combinaison des attributs énumérés ci-dessus.

```
\DeclareLayer[liste d'options]{nom de la couche}
\DeclareNewLayer[liste d'options]{nom de la couche}
\ProvideLayer[liste d'options]{nom de la couche}
\RedeclareLayer[liste d'options]{nom de la couche}
\ModifyLayer[liste d'options]{nom de la couche}
```

Ces commandes peuvent être utilisées pour définir une couche. Le nom de la couche doit être entièrement extensible et s'étendre seulement aux lettres, en caractères ASCII. Certains caractères supplémentaires sont tolérés, mais ne sont pas recommandés. Dans ce cas, si vous utilisez la commande `\DeclareLayer`, elle ne se soucie pas de savoir si une couche avec le même nom existe déjà. La couche sera, dans chaque cas, définie avec les propriétés données dans la liste des options. Une option peut être soit une clé, soit une clé suivie d'un signe égal et une valeur. Plusieurs options peuvent être enchaînées à une liste d'options et elles doivent être séparées par des virgules. Si vous souhaitez utiliser une virgule ou une espace (blanc) à l'intérieur de la valeur d'une option, vous devez mettre cette valeur entre accolades.

Une vue d'ensemble des options et de leurs propriétés est donnée dans la table 16.1 page ci-contre.

Contrairement à `\DeclareLayer`, `\DeclareNewLayer` signale par une erreur si une couche avec le nom spécifié existe déjà. Ainsi, l'utilisateur ne peut utiliser plusieurs fois le même nom, même par inadvertance. Ceci est particulièrement utile lorsque les classes ou paqs internes définissent également les niveaux. Si vous utilisez `\ProvideLayer` au lieu de `\DeclareLayer`, la déclaration sera ignorée dans le cas où une couche existe déjà avec le même nom. Mais si le nom est, cependant, déjà utilisé pour un autre niveau, la définition récente est ignorée. Si une couche existante doit être redéfinie, `\RedeclareLayer` ou `\ModifyLayer` peuvent être utilisés. `\RedeclareLayer` permet tout simplement de définir la couche comme si elle avait été définie récemment. `\ModifyLayer` modifie seulement ces attributs, qui sont représentés par une option de la liste

d'options. Tous les autres attributs resteront inchangés et ne seront pas remis à la valeur initiale par défaut.

Une application qui n'a pas été préalablement définie se traduira par une erreur dans les deux états.

TABLE 16.1 – Options pour la définition des couches de page avec la description de l'attribut correspondant

Éléments
<p><b>addcontent=code</b>            La valeur spécifiée sera ajoutée à la valeur actuelle de l'attribut <b>contents</b>. Il est ainsi généré un contenu supplémentaire. Pour plus d'informations sur le traitement du code de voir option <b>contents</b>.</p>
<p><b>addheight=hauteur supplémentaire</b>            La valeur actuelle de l'attribut <b>height</b> (hauteur) est incrémenté par la valeur de l'option. Cette valeur est la même que <b>height</b> lorsque cela est possible.</p>
<p><b>addhoffset=distance horizontale supplémentaire</b>            La valeur actuelle de l'attribut <b>hoffset</b> est augmentée de la valeur de cette option. Les mêmes attributs que dans <b>hoffset</b> sont possibles comme valeur.</p>
<p><b>addvoffset=espace vertical supplémentaire</b>            La valeur de l'attribut <b>voffset</b> est incrémentée de la valeur de l'option. Les mêmes spécifications que <b>voffset</b> sont possibles comme valeur.</p>
<p><b>addwidth=largeur supplémentaire</b>            La valeur actuelle de l'attribut <b>width</b> (largeur) est incrémentée de la valeur de l'option. Cette valeur est la même que <b>width</b> lorsque cela est possible.</p>
<p><b>align=repère d'alignement</b>            Les repères d'alignement définissent l'alignement souhaité de la couche, chaque repère influence comment sera utilisée la longueur de l'argument des options <b>hoffset</b> ou <b>voffset</b>. Plusieurs repères peuvent être utilisés ensemble (sans virgule ni espace) et interprétés dans l'ordre d'apparition. Les macros ne sont pas autorisées dans la valeur de l'option. Les repères d'alignement admissibles sont :</p>
<p><b>b</b> - alignement de la couche à son bord inférieur ; la valeur de <b>voffset</b> est interprétée comme distance entre bord supérieur du papier et bord inférieur de la couche.</p>
<p><b>c</b> - alignement de la couche en son centre ; les valeurs de <b>voffset</b> et <b>hoffset</b> sont interprétées comme distance entre coin supérieur gauche du papier et centre de la couche.</p>

---

Éléments (suite)

`l` - alignement de la couche à son bord gauche : la valeur de `hoffset` est interprétée comme distance entre bords gauches du papier et de la couche.

`r` - alignement de la couche à son bord droit ; la valeur de `hoffset` est interprétée comme distance entre bord gauche du papier et bord droit de la couche.

`t` - alignement de la couche à son bord supérieur ; la valeur de `voffset` est interprétée comme distance entre bords supérieurs du papier et de la couche.

`area={hoffset}{voffset}{width}{height}`

Propriété composite résultant des primaires `hoffset=distance horizontale`, `voffset=distance verticale`, `width=largeur`, `height=hauteur`.

`backandforeground`

Avec cette option, la limite de niveau à l'avant ou à l'arrière sera annulée. En général, cela a peu d'importance et l'option existe par souci d'exhaustivité. Cette option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`background`

Imprime la couche de fond dans le fond, mais pas dans le premier plan et empêche une double impression. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`bottommargin`

La propriété composite définit `hoffset`, `voffset`, `width`, `height` et `align` le papier du bord gauche vers la droite horizontalement et verticalement, de la zone sous le pied de page vers le bord inférieur du papier. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`clone=nom de la couche composée`

L'option définit les attributs primaires de la couche aux mêmes valeurs que les attributs primaires de la couche avec le nom de couche repère, ce nom de couche doit être entièrement extensible et devrait s'étendre seulement avec des lettres. Certains caractères supplémentaires sont tolérés, mais ne sont pas recommandés. La couche avec le nom d'un niveau donné doit également exister.

`contents=code`

Le `code` spécifié est toujours en expansion et exécuté lorsque la couche est éditée. Ce `code` définit ce qui doit être vu sur le niveau. Il n'y a pas de tests effectués, si le `code` est valide et correct. Donc une erreur dans le `code` peut entraîner plusieurs échecs sur chaque page, qui imprime le `layer`.

`evenpage`

Avec cette option de sortie, le plan à un niveau pour les côtés gauche mais pas pour les pages impaires. Par défaut, l'option n'est pas définie et donc les couches seraient imprimées sur les pages impaires et sur les pages paires.

---

## Éléments (suite)

Puisqu'il y a seulement les pages de gauche dans la disposition à double face, cette propriété comprend un quasi `twoside`. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`everypage`

Combinaison de `odddorevenpage` et `floatornonfloatpage`. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`everyside`

Cette option annule la limite de niveau dans la mise en page unilatérale ou recto-verso qui reprend ses valeurs par défaut. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`floatornonfloatpage`

Avec cette option, la restriction sur le plan des niveaux des côtés coulissants ou non est annulée et le réglage par défaut restauré. Cette option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`floatpage`

Cette option permet à la couche un glissement sur le niveau. L'attribut n'est pas défini par défaut et la couche serait imprimée sur les pages flottant et non-flottant. Noter que l'attribut `subsume twoside.floatpage` imprime la couche seulement sur les pages de flottant, mais pas sur d'autres pages. Il est généré par opposition à la configuration par défaut, il n'est pas sur tous les côtés, mais uniquement sur les cotés de glissement. Plus de détails sur les surfaces de glissement de l'introduction sont donnés à cette `section`. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`foot`

La propriété définit le composite caractéristiques primaires `hoffset`, `voffset`, `width`, `height` et `align` de sorte que le plan recouvre le pied de page dans la largeur de la zone de texte. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`footskip`

La propriété composite définit la caractéristiques principale `hoffset`, `voffset`, `width`, `height` et `align` de sorte que le plan vertical de la gamme soit entre la zone de texte et le pied de page dans la largeur de la zone de texte couverte. (noter que ce n'est pas le même que `\footskip`). L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`foreground`

---

Éléments (suite)

Imprime la couche seulement dans le premier niveau, mais pas dans le fond. Cette couche uniquement de premier plan s'oppose à la couche par défaut qui est à la fois, couche de fond et d'avant-plan et donc serait imprimée deux fois. Par défaut l'attribut n'est pas défini et la couche serait imprimée sur les pages `float` et `non-float`. L'option n'attend pas de valeur.

---

`head`

Définit les caractéristiques primaires `hoffset`, `voffset`, `width`, `height` et `align` pour couvrir le côté de l'entête dans la largeur de la zone de texte.

`headsep`

La propriété définit les caractéristiques primaires `hoffset`, `voffset`, `width`, `height` et `align` pour que le plan couvre la distance entre l'entête de page et la zone de texte dans la largeur de la zone de texte. Leur hauteur correspond à la longueur `\headsep`. N'attend pas de valeur et n'est pas défini par défaut.

---

`height=hauteur`

Définit la hauteur de la couche. Noter que la longueur peut être soit la longueur  $\text{\LaTeX}$ , déclarée en utilisant `\newlength`, soit une longueur de  $\text{\TeX}$ , déclarée en utilisant `\newdimen` ou `\newskip`, une valeur de longueur comme 10 pt, ou une expression de dimensions en utilisant `+`, `-`, `/`, `*`, `(`, et). La syntaxe exacte de l'expression de la longueur est donnée dans [Tea98], section 3.5].

---

`hoffset=distance`

Définit la distance `offset` de la couche (selon `align` pour le bord gauche, le milieu ou le bord droit de la couche) à partir du bord gauche du papier. Comment la distance est mesurée dépend de la propriété l'alignement. Noter que la distance peut être le cas échéant, une longueur  $\text{\LaTeX}$  définie avec `\newlength`, une longueur de  $\text{\TeX}$  définie avec `\newdimen` ou `\newskip`, une longueur d'une valeur de 10 pt ou une expression de longueur en utilisant `+`, `-`, `/`, `*`, `(`, et). La syntaxe d'expression de longueur est donnée dans [Tea98], section 3.5].

---

`innermargin`

La propriété composite `margin` intérieure définit les caractéristiques `hoffset`, `voffset`, `width`, `height` et `align` de manière à couvrir la distance entre le bord droit de la zone de texte et le bord droit du papier sur les pages paires ou la distance entre le bord gauche du papier et le bord gauche de la zone de texte sur les pages impaires et, verticalement, l'ensemble du papier depuis le bord supérieur jusqu'au bord inférieur. L'option n'attend pas de valeur.

---

`leftmargin`

---

## Éléments (suite)

L'option composée définit `hoffset`, `voffset`, `width`, `height` et `align` de manière à couvrir la distance depuis le bord supérieur gauche du papier, le bord gauche de la zone de texte et à étendre verticalement le tout jusqu'au bord inférieur gauche du papier. Non défini par défaut, n'attend pas de valeur.

`mode=modus`

Cette propriété primaire détermine le mode de sortie du niveau. La valeur par défaut est `text`. La ligne de base supérieure est placée à la hauteur d'une ligne standard de texte en dessous du bord supérieur de la couche. Ce texte est en principe correctement aligné sur le haut. En mode `picture`, l'environnement est réglé avec l'origine dans le coin inférieur gauche de la couche. Le mode prédéfini `raw` correspond aussi à `text` par défaut. La modification du mode d'un niveau conduit généralement à un changement du contenu. Le mode `picture` appelle des commandes de positionnement qui conduisent, dans un mode différent, à des messages d'erreur. Par conséquent, il n'est généralement pas souhaitable de modifier, plus tard, le mode de la couche d'un niveau.

`nonfloatpage`

Avec cette option, la couche est limitée à des pages sans flottant. Elle est donc générée par opposition à la configuration par défaut non plus sur tous les côtés, mais seulement sur les cotés non glissants. Par défaut, l'attribut n'est pas défini et les couches seraient imprimées sur les pages `float` et `non-float`.

`oddorevenpage`

Cette option abroge la limite de niveau sur les côtés droit ou gauche, ce qui permet une sortie des deux cotés. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`oddpage`

Cette option permet de gérer une couche à un niveau, côté droit, et d'imprimer la couche seulement sur les pages impaires. Par défaut, l'option n'est pas définie et les couches seront imprimées sur les pages impaires et sur les pages paires. Il convient de noter que toutes les pages sont indépendantes du nombre de pages droites unilatérales. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`oneside`

Avec cette option, le niveau imprime la couche en mode unilatéral seulement, mais pas en mode recto-verso. L'option n'attend pas de valeur.

`outermargin`

---

Éléments (suite)

la propriété composite définit les principales caractéristiques `hoffset`, `voffset`, `width`, `height` et `align` de sorte que la couche couvre la distance entre les bords gauches du papier et de la zone de texte sur les pages paires ou la distance entre les bords droits de la zone de texte et du papier sur les pages impaires et couvre le papier du bord supérieur au bord inférieur.

Cette option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

`page`

La propriété définit le composite caractéristiques primaires `hoffset`, `hoffset`, `voffset`, `width`, `height` et `align` de sorte que le plan couvre la page complète.

L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`pretocontents=code`

La valeur indiquée est conforme à la valeur du contenu actuel de l'attribut. En outre, il s'agit d'un contenu supplémentaire généré jusqu'à maintenant. Pour plus d'informations sur le traitement de `code` voir option de contenu.

-----  
`rightmargin`

La propriété définit le composite caractéristiques primaires `hoffset`, `voffset`, `width`, `height` et `align` de sorte que le plan couvre la droite de la page depuis l'arête supérieure du papier jusqu'au bord inférieur du papier. Cette option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`textarea`

La propriété définit le composite caractéristiques primaires `hoffset`, `voffset`, `width`, `height` et `align` de sorte que la couche couvre la zone complète de texte. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`topmargin`

La propriété composite définit les principales caractéristiques `hoffset`, `voffset`, `width`, `height` et `align` de façon que la couche couvre le papier en démarrant en haut de la page, du bord gauche au bord droit. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`twoside`

Cette option permet l'impression de la couche en recto-verso. L'attribut n'est pas défini par défaut et les couches seront imprimées en mode recto-verso ou recto. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`unrestricted`

Annule toutes les restrictions. Cette option est une combinaison de `backandforeground`, `everyside` et `floatornonfloatpage`. L'option n'attend pas de valeur et l'attribut n'est pas défini par défaut.

-----  
`voffset=distance`  
-----



---

**Éléments (suite)**

Définit la position de la couche au bord supérieur du papier. La façon de mesurer la distance dépend de la propriété `align`. Noter que la distance peut être une longueur  $\text{\LaTeX}$  définie avec `\newlength`, une longueur  $\text{\TeX}$  définie avec `\newdimen` ou `\newskip`, une valeur telle que 10 pt ou une expression de longueur en utilisant `+`, `-`, `/`, `*`, `(`, `et`). Voir la syntaxe exacte de l'expression d'une longueur dans [Tea98], section 3.5.

`width=largeur`

Définit la largeur de la couche. Noter que la distance peut être une longueur  $\text{\LaTeX}$  définie avec `\newlength`, une longueur  $\text{\TeX}$  définie avec `\newdimen` ou `\newskip`, une valeur telle que 10 pt ou une expression de longueur en utilisant `+`, `-`, `/`, `*`, `(`, `et`). Voir la syntaxe exacte de l'expression d'une longueur dans [Tea98], section 3.5].

---

<code>\layerhalign</code>
<code>\layervalign</code>
<code>\layerxoffset</code>
<code>\layeryoffset</code>
<code>\layerwidth</code>
<code>\layerheight</code>

Ces instructions ne sont valables que dans une sortie spécifiée avec le code de `contents`, `addContents` ou `pretocontents`. Dans ce cas, elles renseignent sur l'orientation, la position et la dimension de la couche qui sont utilisées effectivement pour la production. Cependant, elles ne sont pas indispensables à l'expansion réelle du contenu, si, par exemple, la couche n'est pas complètement remplie en largeur ou en hauteur. La propriété de couche primaire s'alignera sur `\layerhalign` et `\layervalign` affichées. Ici, les valeurs horizontales « l » et « r » sont adoptées par `\layerhalign`, tandis que les valeurs verticales « t » et « b » le sont dans `\layervalign`. Les valeurs horizontale et verticale « c » sont transférées simultanément dans les deux directions. Plusieurs informations contradictoires sur `align` peuvent être trouvées avec `\layerhalign` soit « l », « c » ou « r » et avec `\layervalign` soit « r », « c » ou « b ». Une redéfinition des instructions et le changement des valeurs stockées ne sont pas autorisés et conduit à des résultats imprévisibles.

<code>\LenToUnit{longueur}</code>
-----------------------------------

Cette déclaration provient de `eso-pic`, version 2.0f. Elle intègre les valeurs de longueur à des multiples de `\unitlength` et peut donc être utilisée à la place des coordonnées ou des valeurs `\unitlength` dépendantes d'un environnement `picture`. Voir aussi [Nie15] et la déclaration suivante à `\putUR`, `\putLL` et `\putLR`. la déclaration est définie uniquement si elle n'est pas déjà définie, par exemple en chargeant `eso-pic`.

<code>\putUL{contenu}</code>
<code>\putUR{contenu}</code>
<code>\putLL{contenu}</code>
<code>\putLR{contenu}</code>
<code>\putC{contenu}</code>

Ces déclarations peuvent être utilisées en propriété par la couche primaire `contents`, lorsque le niveau a été créé avec `mode=picture` et `\putUL` place le contenu par rapport à l'angle supérieur gauche de la couche, qui correspond à `\put(0,\LenToUnit{\layerheight})`.

Comme `\putUR` correspond aux commandes `\put(\LenToUnit{\layerwidth})` et `\LenToUnit{\layerheight}` son contenu est donc placé par rapport à l'angle supérieur droit de la couche.

`\putLL` place le contenu par rapport à l'angle inférieur gauche de la couche qui correspond à mettre `\put(0.0)`.

`\putLR` place le contenu par rapport au coin en bas à droite du niveau, et correspond à `\put(\LenToUnit{\layerwidth},0)`, et `\putC` place finalement le contenu à mi-couche.

Si vous voulez déterminer avec précision, dans `DIV=classique`, la hauteur du texte correspondante effectivement à la largeur de la page et créer un plan contenant à la fois la zone de texte et un cercle de la largeur du papier dont le diamètre sera placé au centre de cette zone de texte :

```
\documentclass[\cott{DIV=classic}]{scrartcl}
\usepackage{pict2e}
\usepackage{sclayer}
\DeclareNewLayer[%
textarea,background,mode=picture,
contents={
\putLL{\line(1,0){\LenToUnit{\layerwidth}}}%
\putLR{\line(0,1){\LenToUnit{\layerheight}}}%
```

```

\putUR{\line(-1,0){\LenToUnit{\layerwidth}}}%
\putUL{\line(0,-1){\LenToUnit{\layerheight}}}%
\putC{\circle{\LenToUnit{\paperwidth}}}%
}
]{showtextarea}
\DeclareNewPageStyleByLayers{test}{showtextarea}
\pagestyle{test}
\begin{document}
\null
\end{document}

```

Comme vous le voyez, c'est un bon exemple de l'ajustement fait par `typearea` pour une valeur d'un nombre entier de DIV. Plus de détails sur le canon du livre médiéval décrit dans l'exemple ci-dessus figurent dans la [section 2.3](#). Déjà présenté dans l'exemple d'instruction `\DeclareNewPageStyleByLayers`, détaillé seulement dans la [section 17.5](#), il est utilisé ici pour définir un style de page, qui utilise effectivement le niveau nouvellement défini.

`\GetLayerContents{nom de couche}`

Cette déclaration permet de déterminer le contenu actuel d'une couche. Sous réserve de noter que lorsque vous utilisez cette déclaration dans le code des contenus d'attributs `contents`, `addcontents` ou `pretocontents`, une récursivité infinie peut se produire pour accéder au contenu actuel et l'utilisateur responsable se doit d'éviter une telle situation.

`\IfLayerExists{nom}{puis code}{sinon autre code}`

Cette commande peut être utilisée pour exécuter du code à la condition qu'une couche ait déjà été définie. Si la couche existe avec le nom spécifié, le code est alors exécuté, sinon (`else`) un autre code. Noter que cette déclaration ne peut pas vraiment tester si une couche existe.

Au lieu de cela elle utilise des heuristiques qui ne peuvent jamais être de faux négatifs, et dans des cas extrêmes s'avérer être des faux positifs, ce qui indique que quelque chose va mal. Ce pourrait être, par exemple, un avertissement pour utilisation d'un paq incompatible ou quand l'utilisateur fait quelque chose qu'il vaut mieux ne pas faire.

`\DestroyLayer{nom de la couche}`

Cette commande recense l'ensemble des macros correspondant à la couche avec le nom de couche donné à `\relax`, si une couche de ce nom existe, elle ne peut plus être utilisée et sera détruite. Si la couche fait partie d'une page

définie avec le style de page `sclayer`, elle est tout simplement ignorée. Les couches détruites peuvent être redéfinies en utilisant `\DeclareNewLayer` ou `\ProvideLayer`, mais ne peuvent pas être modifiées à l'aide de `\RedeclareLayer` ou `\ModifyLayer` plus longtemps.

Elle est destinée à être utilisée dans `\sclayerOnAutoRemoveInterface`, en interne. Elle permet d'éliminer des niveaux définis à l'aide des macros amovibles d'une interface, chaque fois que l'interface est supprimée.

`\layercontentsmeasure`

La commande `\layercontentsmeasure` est utilisée en interne pour visualiser les couches si l'option `projet` a été définie pour le paq `sclayer`. Dans ce mode de conception, la dimension est sortie en premier après chaque couche. Effectuée avec `\layercontentsmeasure`, sa visualisation se fera avec une règle graduée en centimètres sur le bord supérieur gauche de la couche et un ruban à mesurer en pouces sur son bord inférieur droit.

La commande `\layercontentsmeasure` comporte une option permettant de l'utiliser simplement comme un code unique pour le contenu des propriétés d'un niveau.

## 16.5 Déclaration et gestion des styles de page

Nous connaissons maintenant les niveaux et nous savons comment ils sont définis et gérés. Mais jusqu'à présent, nous ne savons pas comment ils sont exploités. La réponse est peut-être surprenante : en utilisant, simplement, les styles de page. Habituellement, les styles de page sont employés, dans  $\text{\LaTeX}$ , pour la définition de l'entête et du pied de page qui sont imprimés en mode recto-verso pour les pages de droite avec numéro impair ou en mode recto sur toutes les pages. Dans ce mode unilatéral, ils sont utilisés sur tous les côtés.

Exploiter les options `oddpaper` et `evenpaper` permet d'arriver directement à un même niveau avec des résultats comparables. L'entête est imprimé avant le contenu de la page et le pied de page après ce contenu, en correspondance avec les attributs premier plan et fond des options d'une couche. Il paraît plus naturel de définir les styles de page plutôt que des listes de niveaux. Mais au lieu des quatre options (`oddpaper`, `evenpaper`, `background` et `foreground`), tous les attributs de couche listés à la section 17.4, tableau 17.1 peuvent être utilisés.

En raison de ces considérations, `sclayer` offre une forme particulière de style de page, le calque. Il est constitué de niveaux avec l'ajout de crochets (*hooks* en

anglais) qui permettent d'insérer du code complémentaire lors de l'exécution de l'application qui sera ainsi élargie. Les utilisateurs chevronnés connaissent déjà ce concept, par exemple, avec `\AtBeginDocument` (voir [Tea05b]) ou `\BeforeClosingMainAux` (voir chapitre 17.4).

Une deuxième forme de style secondaire fourni par `sclayer` est le style de page `alias`. Il s'agit simplement d'un style différent de la page. En d'autres termes, le nom d'un style de page `alias` est le nom d'`alias` d'un autre style de page, le style `alias` ou la page d'origine.

Pour ce motif, la manipulation de la page de résultats d'un `alias` de style est identique à celle de la page du style original. Si le style de page d'origine est aussi un style de page `alias`, une manipulation de ce style de page `alias` sera à faire, et ainsi de suite jusqu'à la modification du véritable style de page. Tous les styles de page qui ne sont pas `alias` sont de véritables pages de style. Les *alias de style de page* peuvent non seulement être déterminés pour les styles de page définis avec `sclayer`, mais aussi pour tous les autres styles de page.

`\currentpagestyle`

Le paq `sclayer` étend la déclaration `\pagestyle` de  $\text{\LaTeX}$  de sorte que l'effet de `\currentpagestyle` soit défini comme le nom du style de page actif. Il est important de noter que `\thispagestyle` ne change pas `\currentpagestyle`. Mais si vous utilisez `\thispagestyle` le résultat de `\currentpagestyle` peut être modifié dans la fonction de sortie de  $\text{\LaTeX}$ . Toutefois, cela n'a d'effet que si `\currentpagestyle` est utilisé pour protéger cette fonction. Il convient de noter que les styles de page décrits plus loin dans cette section n'attendent pas l'extension `\pagestyle` car également redéfinis par `\currentpagestyle`. Cette extension est faite pour une utilisation avec d'autres styles qui ne sont pas basés sur `sclayer`. Noter également que, après le chargement de `sclayer` et avant le premier appel de `\pagestyle`, `\currentpagestyle` est vide. Si vous définissez une interface utilisateur final du style de page, il peut être utile d'utiliser un `\pagestyle` implicite pour définir la page style actuelle avec un style de page par défaut.

`\BeforeSelectAnyPageStyle{code}`  
`\AfterSelectAnyPageStyle{code}`

La commande `\BeforeSelectAnyPageStyle` permet d'ajouter du code entre crochets qui sera exécuté, immédiatement avant que le style de page ne soit sélectionné, pour l'utilisation de `\pagestyle`. Vous pouvez utiliser `#1` pour noter une espace réservée comme argument de `\pagestyle`. La commande `\AfterSelectAnyPageStyle` fonctionne de façon similaire.

Le code des deux commandes sera exécuté juste après la sélection du style de page et le réglage de `\currentpagestyle` sur le nom du véritable style de page. Ce style doit être impérativement sélectionné à l'aide de `\pagestyle`. S'il l'est d'une manière différente, en utilisant `\thispagestyle` par exemple, le code ne sera pas exécuté.

Noter également : vous ne pouvez pas supprimer le code du crochet après l'avoir ajouté. Toutefois, le code est ajouté localement. Son champ d'application peut donc être restreint à un groupe, pour en limiter la portée.

```
\DeclarePageStyleAlias{nom alias pagestyle}{nom original
pagestyle}
\DeclareNewPageStyleAlias{nom alias pagestyle}{nom original
pagestyle}
\ProvidePageStyleAlias{nom alias pagestyle}{nom original
pagestyle}
\RedeclarePageStyleAlias{nom alias pagestyle}{nom original
pagestyle}
```

Ces commandes peuvent être utilisées pour définir un style de page avec le nom `alias` ou le nom d'un style de page qui est tout simplement un `alias` pour un style de page déjà existant avec un nom original. S'il existe déjà un style portant le nom de la page `alias`, il sera détruit avant de créer l'`alias` utilisant `\DeclarePageStyleAlias` ou `\RedeclarePageStyleAlias`.

La commande `\DeclareNewPageStyleAlias` génère un message d'erreur, si un style de page avec ce nom d'`alias` a déjà été défini auparavant, que cette page de style existante soit elle-même une page `alias`, une page de couche ou un autre type de page n'a aucune importance. De la même manière, la commande `\ProvidePageStyleAlias` définit le style de page d'`alias` uniquement si un style de page avec le nom de la page de style `alias` n'existe pas déjà. Si un tel style de page existe, il est conservé et la déclaration n'a aucun effet.

Contrairement aux autres états, `\RedeclarePageStyleAlias` attend un nom déjà existant de la page `alias` de style de page. Si ce nom n'existe pas, vous obtiendrez un message d'erreur.

```
\DestroyPageStyleAlias{nom pagestyle alias}
```

Avec cette déclaration, la page `alias` de style ne sera pas redéfinie comme nom du style de page d'`alias` spécifié pour L<sup>A</sup>T<sub>E</sub>X si un style de page `alias` de ce nom existe déjà.

Elle sera détruite et l'alias redéfini avec `\DeclareNewAliasPageStyle` ou `\ProvideAliasPageStyle`. Devenue un argument de suppression des styles de page définis pour un utilisateur final, elle utilise les macros amovibles de cette interface à l'intérieur du code de `\sclayerOnAutoRemoveInterface` et annule la suppression automatique de l'interface. (plus de détails chapitre 17.8)

```
\GetRealPageStyle{nom pagestyle}
```

Cette déclaration cherche récursivement le nom réel d'un style de page, si le nom spécifié du style de page appartient à un style de page d'alias. Dans tous les autres cas, même si aucun alias ni aucune page de style n'est nommée, le résultat serait tout simplement le nom de la page de style elle-même. Cela reste vrai, même s'il n'existe pas un style de page appelée `nom pagestyle`.

La déclaration est entièrement extensible et peut être utilisée, par exemple, dans le second argument d'une déclaration `\edef`.

```
\DeclarePageStyleByLayers[listoptions]  
{nompagestyle}{listecouches}  
\DeclareNewPageStyleByLayers[listoptions]  
{nompagestyle}{listecouches}  
\ProvidePageStyleByLayers[listoptions]  
{nompagestyle}{listecouches}  
\RedeclarePageStyleByLayers[listoptions]  
{nompagestyle}{listecouches}
```

Ces déclarations affichent un style de page avec le nom `{nom pagestyle}`. Le style de page se compose d'un certain nombre de niveaux qui sont spécifiés dans la liste des couches séparées par des virgules. Il est important de noter que `nom pagestyle` et liste de couches doivent être totalement extensibles et peuvent intégrer une série de lettres. Certains autres caractères sont tolérés mais ne sont pas recommandés. La liste des options est un listage d'éléments séparés par des virgules, de la forme `clé=valeur`, qui peuvent être utilisés pour définir des propriétés supplémentaires et tirer parti de possibilités accessoires. Ils sont actuellement exploités pour coder certains points d'activation ou définir un style de page entre crochets. Voir l'avant-propos de cette section pour une information plus générale sur les crochets. Des détails sur le crochet et sa signification sont également présentés dans la table 16.2 page suivante ci-après.

TABLE 16.2 – Options entre crochets pour les styles de page (dans l’ordre de leur exécution)

Éléments	Descriptions
<code>onselect=code</code>	Ce code crochet est exécuté lorsque le style de page est sélectionné, par exemple avec <code>\pagestyle</code> . Noter que <code>\thispagestyle</code> ne sélectionne pas le style de page directement, mais qu’il l’active dans la routine d’impression de $\text{\LaTeX}$ .
<code>oninit=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée, deux fois pour chaque page : une fois pour la couche de fond et une fois pour le premier plan.
<code>ononeside=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour le style de page en mode unilatéral, deux fois pour chaque page : une fois pour la couche de fond et une fois pour le premier plan.
<code>ontwoside=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour un style de page dans le mode recto-verso, deux fois pour chaque page : une fois pour la couche de fond et une fois pour le premier plan.
<code>onoddp=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour le style de page sur une page impaire, côté droit, deux fois pour chaque page : une fois pour la couche de fond et une fois pour le premier plan. Noter également que, en mode unilatéral, non seulement les pages avec des numéros impairs, mais toutes les autres pages sont considérées comme pages impaires.



Éléments (suite)	Descriptions
<code>onevenpage=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour le style de page sur une page paire, côté gauche, deux fois, de chaque côté, pour chaque page : une fois pour la couche de fond et une fois pour le premier plan. Noter que, en mode unilatéral, le côté gauche n'existe pas.
<code>onfloatpage=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour le style de page sur un environnement flottant, deux fois, de chaque côté, pour chaque page : une fois pour la couche de fond et une fois pour le premier plan. Noter également qu'une page de flottant est une page qui contient un objet flottant.
<code>onnonfloatpage=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour le style de page sur un environnement non-flottant, deux fois, de chaque côté pour chaque page : une fois pour la couche de fond et une fois pour le premier plan. Noter également que des pages non-flottant sont toutes les pages qui peuvent avoir des <code>t-</code> , <code>b-</code> , ou <code>h-</code> placés ou non... d'objet flottant.
<code>onbackground=code</code>	Le code est exécuté lorsque la sortie des couches du style de page est initialisée pour l'arrière-plan d'une page. Noter que ce sera fait une fois pour chaque page.
<code>onforeground=code</code>	Le code est exécuté lorsque la sortie des couches est initialisée pour le premier plan d'une page. Noter que ce sera fait une fois pour chaque page.

Bien qu'avec la déclaration `\DeclarePageStyleByLayers` le style de page est toujours défini, avec `\DeclareNewPageStyleByLayers` un message d'erreur est émis si un style de page avec le nom de la page de style existe déjà.

Noter que la déclaration d'un style de page qui est déjà un `alias` pour un autre style de la page (voir par exemple `\DeclareAliasPageStyle` plus haut dans cette section) n'entraîne pas la redéfinition du style de page `alias` spécifié,

mais appelle le style de page réel (voir `\GetRealPageStyle` plus avant dans cette section).

En revanche, `\DeclarePageStyleByLayers` et `\ProvidePageStyleByLayers` ne feront rien si un style de page de ce nom existe déjà.

S'il n'existe pas, il peut être redéfini avec `\DeclarePageStyleByLayersindex` ou avec `\DeclareNewPageStyleByLayers`.

La commande `\RedeclarePageStyleByLayers` ne peut être utilisée que si la page de style réelle de nom de page de style existe déjà. S'il n'existe aucune page de style avec le nom spécifié se traduit par un message d'erreur. Noter aussi les commentaires suivants sur le pseudo-style de page `@everystyle@`.

<pre>\pagestyle{@everystyle@} \pagestyle{empty}</pre>
---

Le paq `sclayer` définit déjà, par lui même, deux styles de page d'une manière spéciale.

Le premier est `@everystyle@`.

Ce style de page ne doit jamais être utilisé comme un style de page normal par exemple avec `\pagestyle` ou `\thispagestyle` ou encore comme la cible d'un style de page `alias`, mais les couches et les crochets de style de cette page seront utilisés par tous les autres niveaux de style de page. Donc, ajouter une couche à ce style de page serait similaire à l'ajout de cette couche à tous les autres styles de pages, même vides. Le crochet de `@everystyle@` sera toujours exécuté avant le crochet correspondant et les niveaux avant chacun des niveaux du style de page actif correspondant.

Une différence essentielle : les commandes se rapportant aux niveaux d'un style `\ForEachLayerOfPageStyle`, `\AddLayersToPageStyleBeforeLayer` ou `\AddLayersToPageStyleAfterLayer` changent de niveau dans le style de page `@everystyle@` et sont ignorées si elles sont appliquées à une page différente de style de calque.

Le second, légèrement différent, est un niveau de style de page vide (`empty`). Habituellement, une page de style vide avec ce nom est déjà définie par le noyau de `LATEX` pour être un style de page sans entête ni pied de page. Dans le cas présent, ce style de page a un entête et un pied de page vides, et le paq de `sclayer` l'assimile à un style de page de niveau sans couche qui peut, néanmoins, être utilisé comme n'importe quel autre style de page de niveau. Le principal avantage de ce style de page par rapport au style de page d'origine de

L<sup>A</sup>T<sub>E</sub>X est qu'il gère les crochets et exécute également les niveaux du pseudo-style de page `@everystyle@`.

Pour chacun des crochets de la table 16.2 page 490 existe aussi une option KOMA-Script. Les noms des options de KOMA-Script sont semblables aux noms des options pour les commandes de déclaration de niveau du style de page.

```
onpsselect=code
onpsinit=code
onpsoneside=code
onpstwoside=code
onpsoddpag=code
onpsevenpag=code
onpsfloatpag=code
onpsnonfloatpag=code
onpsbackground=code
onpsforeground=code
```

Un « ps » est inséré après le « on » au début du nom. Les valeurs de ces options de KOMA-Script sont utilisées comme valeurs initiales pour les crochets appropriés. Cette valeur initiale est alors étendue à toutes les valeurs qui sont attribuées aux crochets appropriés dans la liste d'options de la déclaration des commandes.

Dans cette section est expliqué comment la valeur initiale peut être supprimée en utilisant la commande `\ModifyLayerPageStyleOptions`.

```
deactivatepagestylelayers=on-off-valeur (simple interrupteur)
\ForeachLayerOfPageStyle{nom de pagestyle}{code}
\ForeachLayerOfPageStyle*{nom de pagestyle}{code}
```

Tant que l'option KOMA-Script `deactivatepagestylelayers` n'est pas activée, du code arbitraire peut être exécuté avec `\ForeachLayerOfPageStyle` pour chaque niveau du style de page avec le code du style de page. Dans ce code « #1 », un espace est réservé pour le nom de la couche en cours de traitement.

Exemple : vous voulez afficher le nom de tous les niveaux de style de page de `scrheadings` dans une liste séparée par des virgules, vous pouvez le faire avec :

```
\newcommand*\commaatlist{}
\ForeachLayerOfPageStyle{scrheadings}{%
\commaatlist#1\gdef\commaatlist{, }}

```

```
\let\commaatlist\relax
```

L'utilisation de `\gdef` lieu de `\def` dans l'exemple est nécessaire parce que le code est exécuté dans un groupe afin de minimiser les effets secondaires indésirables.

Cependant, la commande `\gdef` définit `\commaatlist` globalement, de sorte que lorsque vous appelez le code pour le niveau suivant, le changement perdure et soit encore valide.

Sinon, on pourrait également agir avec `\def`, mais cela fonctionne aussi avec la version étoilée `\ForEachLayerOfPageStyle*`. Cette forme est omise dans l'exécution de code sur un groupe supplémentaire. L'utilisateur doit s'assurer lui-même que le code n'a pas d'effets secondaires indésirables. En particulier, la désactivation du code avec `deactivatepagestylelayers=true` se ferait alors en appelant `\ForEachLayerOfPageStyle*` qui est aussi supporté.

Les instructions générales de `sclayer` utilisent `\ForEachLayerOfPageStyle` en interne. En outre, l'option de KOMA-Script `deactivatepagestylelayers` permet de changer la fonction.

Cette option peut donc être utilisée pour désactiver temporairement tous les plans de tous les styles de page ou pour cacher toutes les couches de tous les styles de page de la couche.

```
\AddLayersToPageStyle{nom de pagestyle}{liste de couches}  
\AddLayersAtBeginOfPageStyle{nom de pagestyle}{liste de couches}  
\AddLayersAtEndOfPageStyle{nom de pagestyle}{liste de couches}  
\RemoveLayersFromPageStyle{nom de pagestyle}{liste de couches}
```

Vous pouvez utiliser ces commandes pour ajouter des couches de calques à une page style ou pour enlever des couches d'un style de page.

Le style de page est ici désigné par son nom de style de page. Les niveaux sont indiqués dans une liste de couches séparées par des virgules. Les commandes `\AddLayersToPageStyle` et `\AddLayersAtBeginOfPageStyle` ajoutent tous les niveaux de la liste des couches séparées par des virgules à la fin de la liste des calques du style de page.

Logiquement, viennent en premier les niveaux nouvellement ajoutés, et ainsi de suite avant les niveaux existants, au-dessus ou en face des anciennes couches de style de la page. Néanmoins, de nouvelles couches de fond seraient derrière le calque de texte et donc derrière toutes les couches de premier plan.

La commande `\AddLayersAtBeginOfPageStyle` ajoute les nouvelles couches au début de la liste des calques du style de page. Donc, la dernière couche de la liste deviendra la nouvelle première couche de cette même liste. Repérer que les couches seront ajoutées dans l'ordre de la liste des calques, la première couche en premier, la deuxième couche en second et ainsi de suite.

La commande `\RemoveLayersFromPageStyle` peut être utilisée pour enlever des couches de la liste des couches au lieu de les ajouter. Repérer que les couches qui font partie de cette liste mais ne font pas partie de la liste des calques du style de page, seront ignorées, sans provoquer de message d'erreur.

```
\AddLayersToPageStyleBeforeLayer{pagestylenom}{liste de couches}
{nom du plan de référence}
\AddLayersToPageStyleAfterLayer{pagestylenom}{liste de couches}
{nom du plan de référence}
```

Ces deux commandes sont similaires aux commandes précédentes, mais elles n'ajoutent pas les couches au début ou à la fin de la liste des couches d'une page de style calque, mais juste avant ou après une couche de référence de cette même liste. Un nom de niveau de référence est donc spécifié, qui sera recherché pour ce plan de référence.

Si le nom de la couche de référence ne fait pas partie de la liste des couches du style de page, rien ne sera inséré. Si, dans le style de page, il n'existe aucun style de page de niveau ni aucun style de page `alias` qui mène à un style de page de niveau, une erreur est signalée.

```
\UnifyLayersAtPageStyle{pagestylenom}
```

Avec les commandes décrites précédemment dans cette section, vous pouvez ajouter des couches différentes à un style de page, mais aussi ajouter une seule couche à plusieurs reprises pour un style de page. C'est donc acceptable, même si, dans la plupart des cas, avoir une couche identique à plusieurs reprises dans la liste des calques d'un style de page de la couche n'a aucun sens. Dans ce cas, vous pouvez utiliser `\UnifyLayersAtPageStyle` pour supprimer les doublons de couche de cette liste.

Il convient de noter que cette action peut changer l'ordre des couches.

Donc pour créer une commande spéciale, vous devez d'abord supprimer toutes les couches, puis ajouter les couches dans l'ordre que vous souhaitez sans utiliser `\UnifyLayersAtPageStyle` n'est pas adapté dans un tel cas.

```
\ModifyLayerPageStyleOptions{nom de pagestyle}{liste d'options}
\AddToLayerPageStyleOptions{nom de pagestyle}{liste d'options}
```

Avec ces deux déclarations les options et les crochets du niveau d'un style de page peuvent être modifiés ultérieurement. Repérer que le réglage d'une nouvelle valeur d'option en utilisant `\ModifyLayerPageStyleOptions` va supprimer la valeur précédente, y compris la valeur globale par défaut. Les options, qui ne figurent pas dans la liste d'options resteront inchangées.

Si vous voulez définir une option « ne rien faire », vous pouvez utiliser la valeur `\relax`.

Seules les options d'une liste d'options de `\ModifyLayerPageStyleOptions`, séparées par des virgules, seront réglées sur les nouvelles valeurs indiquées dans la liste des options si cette nouvelle valeur n'est pas vide, mais attention, les valeurs précédentes seront perdues. Ceci est applicable à toutes les options de la table 16.2 page 490.

Les options ou crochets qui ne sont pas répertoriés dans la liste des options restent inchangés. Cette déclaration est donc le seul moyen de supprimer les préférences globales des options d'un style de page KOMA-Script.

La commande `\AddToLayerPageStyleOptions` diffère dans la mesure où les nouvelles valeurs n'écrasent pas les valeurs précédemment existantes, mais sont ajoutées ou - plus précisément - concaténées aux valeurs présentes.

La déclaration de `\AddToLayerPageStyleOptions`, toutefois, ne remplace pas les valeurs qui existaient auparavant, mais ajoute les nouvelles à ce jour - plus précisément - accrochent les nouvelles valeurs aux anciennes.

```
\IfLayerPageStyleExists{nom de pagestyle}{code}{code else}
\IfRealLayerPageStyleExists{nom de pagestyle}{code}{code else}
```

`\IfLayerPageStyleExists` teste si la page réelle du style de page est, oui ou non, un style de page de la couche ou le nom d'un style de page d'alias. Si c'est le cas et si le test est vrai (true), alors le code est exécuté. En interne, cette commande est souvent utilisée pour lancer un message d'erreur si vous utilisez l'une des commandes couche de style de page avec un nom de page de style qui ne correspond pas à une page de style de la couche. Si le nom du style de page n'est ni un style de page de calque, ni un alias d'un style de page de calque, ni un alias d'un alias de... un style de page de calque, le code else sera exécuté.

La commande `\IfRealLayerPageStyleExists` va encore plus loin et dirige le code seulement si le nom lui-même, indiqué sur la page style de page style,

est le nom d'un style de page de niveau. Donc, le code `else` sera exécuté, même si le nom du style de page est un nom d'`alias` d'un style de page de couche ou le nom d'`alias` d'un nom d'`alias` de... un style de page de couche.

```
\IfLayerAtPageStyle{nom de pagestyle}{niveau de nom}{code} {code
else}
\IfSomeLayersAtPageStyle{nom de pagestyle}{niveau de liste}
{code}{code else}
\IfLayersAtPageStyle{nom de pagestyle}{niveau de liste} {code}{code
else}
```

Ces commandes permettent de vérifier si un ou plusieurs niveaux font partie d'un style de page. La commande `\IfLayerAtPageStyle` attend le nom du style de page comme premier argument et le nom d'un niveau comme second argument. Si le test est vrai (`true`), le code sera alors exécuté. Si la couche n'est pas dans la liste, le code `else` sera exécuté.

Avec l'arbitrage de `\IfSomeLayersAtPageStyle` et `\IfLayersAtPageStyle` pour permettre d'utiliser, dans le second argument, une liste de couches séparées par des virgules. `\IfSomeLayerAtPageStyle` exécute le code si au moins l'une des couches de la liste figure sur la liste de noms de couches de style de page. En revanche, `\IfLayersAtPageStyle` ne l'exécute que si tous les niveaux font partie du style de page. Si la condition n'est pas remplie, le code `else` est réalisé.

Cependant des conditions complexes de nidification entraînent des restrictions et peuvent être signalées. On donne à la place d'une liste de couche seulement le nom du niveau, de sorte que les trois déclarations aient une signification équivalente.

```
\DestroyRealLayerPageStyle{nom du niveau d'un style de page}
```

Avec cette instruction on peut redéfinir un niveau de la page de style de  $\text{\LaTeX}$ , si, et seulement si, c'est une page de style de la couche. Rien ne se passerait si ce nom est un nom d'`alias` d'une couche de style de page, d'un autre style de page, ou si ce n'est pas un nom de style de page et la déclaration serait ignorée.

Au cas où le nom de couche du style de page est le nom actuel du style de page, le style de page actuel deviendra une sorte de style de page vide.

Au cas où le style particulier de la page fixé par `\thispagestyle`, est au niveau du nom de style de page, il vient d'être réinitialisé et tout simplement remis à zéro.

La précédente commande `\thispagestyle` ne sera plus valable. Vérifier que les couches du style de page ne sont pas automatiquement détruites. Si vous voulez aussi les supprimer, vous pouvez le faire par exemple avec :

```
\ForEachLayerOfPageStyle{...}{\DestroyLayer}{#1}
```

avant de détruire la couche du style de page.

La déclaration est destinée à être utilisée pour les styles de page définis dans le cadre d'une des interfaces utilisateur final, avec la suppression automatique de cette interface dans le code de l'argument `\sclayerOnAutoRemoveInterface` (Voir l'article 17.8)

## 16.6 Hauteurs de l'entête et du pied de page

L'entête et le pied de page sont des éléments clés de l'empage. En outre, la définition des options d'un niveau peut être particulièrement limitée (voir la table 16.1 page 477). Par conséquent, la hauteur de ces deux éléments doit être connue. Ce qui est écrit dans la section 5.2 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 5.2, vous pouvez continuer à l'article 17.7.

<code>\footheight</code>
<code>\headheight</code>

Le paq `sclayer` amène une nouvelle longueur `footheight` analogue à la hauteur `headheight` du noyau  $\text{\LaTeX}$ . Simultanément `footskip` est interprété par `\sclayer` comme étant la distance de la ligne de base finale de la zone de texte à la première ligne de base standard du pied. Le paq `typearea` voit cela de la même manière, de sorte que les options locales pour le réglage de la hauteur du pied (voir les options `footheight` et `footlines` dans la section 2.6) tiennent compte de ce pied dans le calcul de la disposition du texte (voir l'option `footinclude` dans la même section). Elles peuvent très bien être utilisées pour définir les valeurs de `sclayer` et également fournir le résultat souhaité.

Hors l'utilisation du paq `typearea`, vous devrez peut-être ajuster la hauteur de l'entête et du pied avec une valeur de paramètre conforme au paq `geometry`. Si la teneur réelle utilisée est trop petite, essayer les longueurs adaptées dans l'exemple `sclayer-scrpage`. Dans le même temps, un avertissement est émis, qui comprend également des conseils pour effectuer les réglages appropriés. Les modifications automatiques sont alors réalisées à partir du moment où leur



nécessité a été reconnue valide et ne sont pas automatiquement annulées lorsque, par exemple, le contenu de l'entête ou du pied devient encore plus petit.

## 16.7 Influence des styles de page définis

Une partie du présent paragraphe fait l'objet d'une étude préliminaire créée, quasiment dans les mêmes termes, au chapitre « Personnaliser le texte ».

Bien que `scrlayer` ne définisse pas, lui-même, un style de page spécifique avec du contenu, le style de page `@everystyle@` déjà mentionné est vide et tous les styles sont initialement définis sans premier niveau ni contenu, mais il fournit quelques options et commandes pour la manipulation du contenu.

```
\automark[niveau hiérarchique de la marge droite]{niveau
hiérarchique de la marge gauche}
\automark[niveau hiérarchique de la marge droite]{niveau
hiérarchique de la marge gauche}
\manualmark
automark
autooneside=on-of-valeur
manualmark
```

Dans la plupart des classes et des rubriques, le choix du style de page `headings` et `myheadings` détermine si les titres sont générés en manuel ou automatique. Cette discrimination a été abolie avec `scrpage2` et maintenant avec `scrlayer`. Au lieu de faire la distinction entre les rubriques automatiques et manuelles selon le style de page, deux commandes `\automark` et `\manualmark` ont été créées.

Avec `manualmark` la commutation est manuelle, l'automatique est désactivé. D'autre part, les niveaux de plan à utiliser pour le réglage automatique peuvent être définis avec `automark` et `automark*`. L'argument optionnel dans ce cas indique le niveau de structuration de la marge de droite, tandis que l'argument obligatoire décrit le niveau de la marge gauche. Dans chaque cas, le nom du niveau du document est spécifié comme argument, à savoir `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` ou `subparagraph`.

Normalement, le niveau supérieur doit définir la marge de gauche, tandis que le niveau inférieur doit être utilisé pour la marge de droite. Bien qu'elles ne soient pas obligatoires, ces conventions communes sont simplement utiles.

Il convient de noter que toutes les classes ne permettent pas à `Kolumnentitel` d'agir à tous les niveaux, par exemple les classes `standard` `Kolumnentitel` `part` ne seront jamais définies alors que les classes KOMA-Script soutiennent tous les niveaux.

La différence entre `automark` et `automark*` est que `automark` annule toutes les commandes précédentes pour régler automatiquement les marges, tandis que la version étoilée `automark*` modifie uniquement les actions pour les niveaux hiérarchiques spécifiés. On peut donc couvrir également les cas relativement complexes.

Outre les commandes, les deux options `manualmark` et `automark` permettent de basculer directement entre les titres automatique et manuel d'exécution en cours, `automark` est toujours utilisé dans les classes avec la déclaration `chapter` comme valeur par défaut, `\automark[section]{chapter}`.

Et avec d'autres classes `\automark[subsection]{section}`.

En principe, dans le mode unilatéral, on ne désire généralement pas que le niveau inférieur affecte la marge de droite. C'est plutôt le niveau supérieur qui indique également la valeur par défaut et qui influe dans l'exemple en mode recto-verso seulement sur la marge gauche pour afficher le titre de toutes les pages en cours. Cette préférence correspond à une option active `autooneside`. Elle peut prendre les valeurs pour les interrupteurs simples qui sont présentées dans le tableau 2.5 dans la table 2.5 page 59. Si l'option est désactivée, elle agit en mode unilatéral comme paramètre obligatoire dans le titre des colonnes.

Le chargement du paq lui-même n'a pas d'effet sur l'opportunité de travailler avec des entêtes de colonnes au fonctionnement automatique ou non. Seule l'utilisation explicite des options `automark` ou `manualmark` ou de l'une des deux commandes `\automark` ou `\manualmark` crée ici des conditions claires.

`draft=on-off-valeur(simple interrupteur)`

L'option KOMA-Script peut prendre les valeurs `standard` pour interrupteur simple, comme celles qui sont présentées dans la table 1.6 page 20. Si cette option est activée, tous les éléments des styles de pages sont en outre fournis avec des lignes de cotes, ce qui peut parfois être utile pendant la phase de conception et pour tout projet de modification.

`\MakeMarkcase{text}`

Le titre courant automatique utilise `\MakeMarkcase` pour ses productions, mais pas le titre courant manuel. Si l'instruction lors du chargement de `sclayer`

n'est pas définie, elle prendra la valeur par défaut de sortie de la chaîne d'argument sans modification.

Cependant, ce pré-réglage peut être modifié en redéfinissant `\MakeMarkcase` ou par l'option `markcase` documentée à la [section 5.5](#), et l'argument est ensuite converti, selon le réglage, en majuscules ou minuscules par exemple.

```
\markcase=valeur
```

Comme expliqué plus haut, vous pouvez choisir avec `scrlayer` entre manuel et automatique pour les titres en cours d'exécution, avec la mise en forme automatique dans l'entête tandis que les marges appropriées seront définies avec les commandes de sectionnement.

Chaque culture obéit à ses propres règles de typographie, parfois différentes de celles appliquées dans les pays germanophones, telles le titre des rubriques en lettres majuscules.

Les classes `standard` le permettent par défaut. Le paq `scrlayer` le supporte également en option, avec la marque facultative `markcase=upper`. En fin de compte, cela conduit à une redéfinition de `\MakeMarkcase`, déclaration utilisée par `scrlayer` dans la mise en place du titre de l'entête en cours d'exécution.

Malheureusement cette adaptation proposée de type `\MakeUppercase` conduit dans  $\text{\LaTeX}$  à une phrase très insuffisante, car non verrouillée ou compensée, sans doute dûe en partie à l'interprétation typographiquement correcte nécessitant une analyse de glyphe afin d'incorporer la forme spécifique des lettres et leurs combinaisons dans la balance de fermeture. Par conséquent, l'auteur de `KOMA-Script` recommande de renoncer à cette application pour `Kolumnentitel`. Ceci est habituellement utilisé avec `markcase=used` possible. Cependant, certaines classes ajoutent `MakeUppercase` ou même la commande  $\text{\TeX}$  `\uppercase`. Pour ces cas, il y a aussi le réglage de `markcase=noupper` qui peut être désactivé localement pour `Kolumnentitel` à l'aide de `\MakeUppercase` et `\uppercase`.

Toutes les valeurs possibles pour `markcase` sont résumées dans la [table 4.2](#) page 266.

```
\leftmark
\rightmark
\headmark
\pagemark
```

Pour déroger aux styles prédéfinis, il faut généralement être en mesure de décider où les repères doivent être placés. Avec `\leftmark` le numéro est à

gauche. Il est ensuite remplacé lors de la sortie de la page avec le contenu approprié.

Par conséquent, vous pouvez utiliser `\rightmark` pour placer le numéro qui sera ensuite remplacé par le contenu correspondant à la sortie de la page, à droite. Pour plus de subtilités, consulter les explications supplémentaires de l'article 21.1.

Utiliser `\headmark` peut vous rendre la vie plus facile. Cette extension de `sclayer` correspond à `\leftmark` ou `\rightmark` selon que la page en cours est impaire ou paire.

La commande `\pagemark` n'a strictement rien à voir avec les sections de `TEX`. Elle sert à formater le numéro de page. Pour créer ce numéro et pour le réglage de l'élément de la police, c'est `pagenumber` qui est utilisé. Cela peut être modifié en utilisant les commandes `\setkomafont` et `\addtokomafont` (voir également la section 3.6)

Dans la section 5.5, vous trouverez un exemple d'utilisation des commandes `\headmark` et `\pagemark` avec `sclayer` basé sur le paq `sclayer-scrpage`.

Si les possibilités offertes pour les marques ne suffisent pas, des instructions sont documentées dans ce chapitre pour les utilisateurs avancés, comme par exemple l'usage de `\leftfirstmark` et de `\rightbotmark` expliqué pour les documents de type encyclopédie.

```
\partmarkformat
\chaptermarkformat
\sectionmarkformat
\subsectionmarkformat
\subsubsectionmarkformat
\paragraphmarkformat
\subparagraphmarkformat
```

Ces instructions sont utilisées en interne par les classes `KOMA-Script` et par `sclayer` pour formater le numéro du titre de la section en cours d'exécution automatique. Ici aussi, le mécanisme `\autodot` des classes `KOMA-Script` est pris en charge. Si nécessaire, ces instructions peuvent être redéfinies afin de parvenir à une autre mise en forme des numéros, voir éventuellement l'exemple dans la section 5.5.

```

\partmark{texte}
\chaptermark{texte}
\sectionmark{texte}
\subsectionmark{texte}
\subsubsectionmark{texte}
\paragraphmark{texte}
\subparagraphmark{texte}

```

Ces instructions sont utilisées en interne par la plupart des classes pour définir les repères en fonction des commandes de sectionnement. Ici, seul le texte, et non un nombre, est attendu comme argument. Le nombre est déterminé automatiquement à la place lors de la lecture des compteurs lorsque l'on travaille avec des titres numérotés.

Cependant une telle instruction n'est pas utilisée dans tous les niveaux de détail de toutes les classes, et `\partmark`, par exemple, n'est jamais appelé par les classes `standard`, alors que les classes `KOMA-Script` bien sûr le soutiennent.

Si ces instructions sont redéfinies par l'utilisateur, il doit être absolument sûr avant de fixer le nombre à vérifier également sur `secnumdepth` que les numéros peuvent être émis.

Ceci est également valable si l'utilisateur ne change pas lui-même `secnumdepth` parce que les paqs et les classes peuvent compter sur l'effet de `secnumdepth`.

Le paquet `sclayer` redéfinit également ces instructions à chaque appel de `\automark` ou `\manualmark` ou des options correspondantes sont renouvelées en partie pour atteindre les positions souhaitées de fonctionnement automatique ou manuel.

```

\markleft{marque gauche}
\markright{marque droite}
\markboth{marque gauche}{marque droite}

```

Peu importe si les gens travaillent actuellement avec des titres manuels ou automatiques en cours d'exécution, vous pouvez définir avec ces instructions et à tout moment, la marge à gauche ou à droite. Il convient de noter que le dernier résultat de la marge gauche dans `\leftmark` est sur la page correspondante, alors que `\rightmark` le premier résultat sort sur la page de droite pertinente. Pour plus de détails, se référer à d'autres explications dans l'article 21.1.

Lorsque vous travaillez avec des titres manuels, les marques restent valables jusqu'à leur remplacement explicite par l'utilisation d'un énoncé correspondant.

Lors de l'exécution d'une configuration automatique des entêtes, elles peuvent perdre leur validité avec une structuration ultérieure.

Toujours dans le contexte en relation avec les commandes de sectionnement versions étoilées, ces instructions peuvent être utiles. Des exemples détaillés de l'utilisation de `\markboth` avec le paq `sclayer` dérivé de `sclayer-scrpage` peuvent être trouvés dans la section 5.5.

`\GenericMarkFormat{nom du niveau de la section}`

Cette commande est utilisée pour le formatage par défaut de tous les numéros de paragraphes dans la gestion des entêtes de sections et de classes inférieures sans `\chapter` et pour le niveau de sections et sous-sections, dans la mesure où aucune déclaration `mark` correspondante n'a été définie ailleurs, par exemple avant le chargement de `sclayer`.

Ainsi le paq emploie par défaut `\@secntmarkformat` si une telle commande n'a pas été définie, comme les classes de `KOMA-Script` le font habituellement, ou la commande `\@secntformat`, fournie par `LATEX`, sera utilisée, modifiée à l'aide de `KOMA-Script`. Comme argument `\GenericMarkFormat` attend un nom de niveau, par exemple un chapitre ou une section sans la barre controblic<sup>2</sup> (`\`) avant.

Vous pouvez modifier, dans les titres en cours d'exécution qui en dépendent, le formatage par défaut des numéros qui l'utilisent en redéfinissant uniquement cette commande. Les classes peuvent également changer la mise en forme par défaut, de la même manière

Un exemple détaillé de l'interaction d'une commande `\GenericMarkFormat` explique les commandes `\sectionmarkformat` et `\subsectionmarkformat` ou `\chaptermarkformat` lorsque vous utilisez le paq `sclayer-scrpage` dérivé de `sclayer` décrit au paragraphe 18.1.

`LATEX` utilise pour le style général de page une marque de `TEX` en deux parties. Dans la partie gauche le titre de la colonne peut être ajouté avec `\leftmark`, tandis que la partie droite avec `\rightmark` reste disponible. En fait, il est probablement aussi prévu que `\leftmark` soit utilisé pour la page de gauche et `\rightmark` pour la page de droite dans une impression recto-verso. Cependant dans la mise en page unilatérale d'un document la commande structurante des classes `standard` utilise la marque de la partie gauche.

---

2. Voir la note de la page xxv.

<code>\righttopmark</code>
<code>\rightbotmark</code>
<code>\rightfirstmark</code>
<code>\lefttopmark</code>
<code>\leftbotmark</code>
<code>\leftfirstmark</code>

$\TeX$  connaît trois façons d'accéder à une marque.

`\botmark` qui est l'ultime marque valable de la dernière page compilée. Cela correspond à la dernière série de la marque de page. Si aucune marque n'a été mise sur le côté, `botmark` correspond à la dernière marque de série sur les pages déjà émises. Les commandes `\leftmark` et `\leftbotmark` utilisent cette marque pour la partie gauche de la note finale et `\rightbotmark` pour la partie droite.

`\firstmark` est la première marque de page récemment compilée et c'est la première marque mise sur la page. Si aucune marque n'a été mise sur le côté, `firstmark` correspond à la première marque de série sur les pages déjà émises. Les commandes `\rightmark`, et `\rightbotmark` utilisent cette marque, et la commande `\firstmark` concernera alors la partie gauche.

`\topmark` est le contenu créé par `\botmark` avant la page courante, et  $\LaTeX$  ne l'utilise jamais automatiquement. Néanmoins `scrlayer` permet l'utilisation de `\lefttopmark` sur la partie gauche de la même marque et d'accéder à la partie droite avec `\righttopmark`.

Noter que la gauche et la droite de la marque peuvent être réglées seulement en commun. Modifier la partie droite avec `\markright` règle simultanément la partie gauche (inchangée). Par exemple, utiliser `\chaptermark` puis `\markboth` avec un argument vide à droite. C'est aussi la raison pour laquelle `\rightmark` ou `\rightfirstmark` sur le côté droit d'un titre de chapitre renvoie toujours une valeur de retour vide, même si, après `\sectionmark` ou indirectement après `\section`, une nouvelle partie droite a été fixée.

Il est bon de savoir que l'utilisation des commandes déclarées ici, sortie gauche ou droite dans une page, peut conduire à des résultats inattendus. Elles sont vraiment destinées à être utilisées dans l'entête ou le pied de page d'un style de page. Elles devraient toujours, par conséquent, être une partie du contenu d'un plan `scrlayer`. Qu'elles soient limitées à l'arrière-plan ou au premier plan n'a, d'autre part, aucune importance puisque tous les niveaux ne seront délivrés qu'après la compilation de la page courante.

Plus de détails sur le mécanisme `mark` de `TEX` peuvent être vus, chapitre 23 [Knu90]. Le thème est traité là avec beaucoup de compétences, mais si vous trouvez les explications plutôt confuses, s'il vous plaît, surtout ne faites rien.

```
\@mkleft{repère gauche}  
\@mkright{repère droit}  
\@mkdouble{repère}  
\@mkboth{repère gauche}{repère droit}
```

Dans les classes et paqs, il est possible que les titres courants d'entête ne soient remplis que si l'option de fonctionnement automatique est activée (voir l'option `automark` et la commande `\automark`). Les classes standard `LATEX` utilisent alors l'option `\@mkboth`. Cette commande est soit `\@gobbletwo`, qui détruit les deux arguments obligatoires, ou `\markboth`, une commande qui remplit à la fois la gauche, et de la gauche vers la droite. Un paq comme `babel` change aussi `\@mkboth`, par exemple commuter la langue dans les rubriques. Mais si un auteur de classe ou de paq ne veut changer qu'un repère ou le repère gauche sans changer le droit, la commande appropriée est manquante. Le paq `sclayer` lui-même exige des instructions *ad-hoc*, par exemple dans le cadre de la mise en œuvre automatisée du titre courant. Donc, si la commande `\@mkleft` est utilisée seulement pour régler le repère gauche, `\@mkright` le repère de droite ou `\@mkdouble` pour régler à la fois les repères droit et gauche, le contenu ne sera pas défini lors du chargement de `sclayer`, mais par le paq lui-même. Ainsi la définition même de `\@mkboth` sera utilisée comme indicateur de l'utilisation des entêtes en fonctionnement automatique, et dans ce cas, il est nécessaire de régler les commandes avec un repère correspondant.

Les auteurs de classes et de paqs peuvent également utiliser ces commandes s'ils veulent remplir un seul des repères, ou les deux, avec le même contenu, mais seulement si la mise en œuvre automatisée du titre courant est activée. Cette condition est aussi la différence entre les commandes `\markleft`, `\markright` et `\markboth`. Pour plus d'informations sur la manipulation du contenu de style de page voir également la section 6.5.

## 16.8 Interfaces pour l'utilisateur final

Le paq `sclayer` fournit une interface utilisateur compétitive pour définir et gérer les interfaces des utilisateurs finaux. Il est possible que, dans le futur, cette interface disparaisse de `sclayer` pour être transférée dans `scrbase`.



Actuellement, cette interface est expérimentale et le paq `scrlayer` fournit ses propres commandes de définition d'interface. Ce serait bien que l'utilisateur ne dépende pas, finalement, du fonctionnement de l'auto-suppression d'un lien expérimental dont, de toute façon, vous devriez éviter l'utilisation.

Cette section décrit uniquement les instructions de liaison pour la définition d'une interface utilisateur final, instructions qui n'intéressent pas les utilisateurs finaux eux-mêmes, mais seulement les auteurs d'interfaces orientées vers les paqs et les classes qui s'appuient sur `scrlayer`. L'utilisateur final trouvera des informations spécifiques sur les interfaces dans les chapitres 5, 18 et 19.

```
\scrlayerInitInterface[nom de l'interface]
```

Cette déclaration enregistre une nouvelle interface titrée *nom de l'interface*, commande obligatoire imposée pour interfaces avec nom d'argument unique. Cela signifie qu'une interface ne doit pas être déjà enregistrée avec le même nom. Si tel est le cas, un message signalant une erreur est renvoyé. Elle doit toujours être la première commande d'une interface pour utilisateur final. Par conséquent, elle est également décrite en premier.

Si l'argument optionnel - y compris les crochets – est omis, alors `\@currname` et `\@currentx` seront utilisés à la place. Pour les classes et les paqs c'est le nom du fichier de la classe ou du paq lors de leur chargement. Mais vous pouvez utiliser n'importe quelle séquence de caractères avec `letter` ou `other`. C'est utile si, par exemple, une classe ou un paq définit plusieurs interfaces pour l'utilisateur final.

```
forceoverwrite=on-off(valeur)
autoremovinterfaces=on-off(valeur)
\scrlayerAddToInterface[nom d'interface]{commande}{code}
\scrlayerAddCsToInterface[nom d'interface]{séquence de
commandes}{code}
```

Une des caractéristiques uniques des interfaces utilisateur final de `scrlayer` est qu'elle devrait enregistrer les commandes utilisées (appelées aussi macros). Et cela peut être fait avec `\scrlayerAddToInterface`. Le nom optionnel de l'argument de l'interface est le nom enregistré avec `\scrlayerInitInterface` auparavant. Les instructions ne sont pas seulement définies lors du chargement d'une classe ou d'un paq, mais le sont également lors de l'exécution, de sorte que l'argument optionnel est également utilisé s'il correspond au nom de fichier de la classe ou du paq parce que les valeurs de `\@currname` et de `\@currentx` sont valides uniquement pendant le chargement.

Une erreur se produit si l'interface n'a pas été initialisée avant.

Le premier argument obligatoire est la commande<sup>3</sup> à ajouter à l'interface de l'utilisateur final. Si la commande peut être ajoutée à l'interface, cela est fait. En outre, la commande sera réglée sur `\relax` et le code exécuté. Vous pouvez utiliser, par exemple, `\newcommand` à l'intérieur du code pour définir la commande.

Mais quand une commande peut-elle être définie ?

Si une commande est indéfinie ou `\relax`, elle peut toujours être définie. Si une commande, toujours définie, a été enregistrée pour une interface utilisateur différente supprimée automatiquement par `KOMA-Script` lors de l'activation de `autoremoveinterface`, régler la commande `\relax` et enregistrer la nouvelle interface utilisateur spécifiée.

Si une commande est déjà définie, mais ne fait pas partie d'une interface et si l'option `forceoverwrite` de `KOMA-Script` a été activée, l'instruction sera réglée sur `\relax` et enregistrée pour l'interface donnée. La commande est donc définie dans ce cas également.

Dans tous les autres cas, elle ne peut pas être définie, en particulier, si elle est déjà définie et que les options `KOMA-Script` `autoremoveinterface` et `forceoverwrite` sont désactivées.

La commande `\sclayerAddCsToInterface` fonctionne de façon similaire à la commande `\sclayerAddToInterface`. Toutefois, elle n'attend pas, comme premier argument, une commande, mais une séquence de commandes<sup>4</sup>.

`\sclayerOnAutoRemoveInterface[nom d'interface]{code}`

Si l'interface utilisateur final appelée « nom d'interface » est automatiquement supprimée (voir `KOMA-Script`, possibilité `autoremoveinterface` au début de cette section) le code peut être également exécuté. Ceci peut par exemple être utilisé pour détruire des couches ou des styles de page (voir `\DestroyLayer`, `\DestroyAliasPageStyle`, et `\DestroyRealLayerPageStyle`) sur la base de commandes de l'interface de l'utilisateur final, automatiquement.

La déclaration `\sclayerInitInterface` est utilisée pour le nom d'argument optionnel de l'interface.

---

3. Une commande se compose d'un controblic (barre oblique inverse, en anglais *backslash*) suivi d'une séquence (de commandes) constituée soit de caractères alphabétiques, soit d'autres signes, ou encore (sans controblic) de caractères actifs.

4. Une séquence de commandes doit être entièrement extensible et l'expansion doit indiquer entre autre la catégorie de lettre et/ou gérer l'espace.

# Chapitre 17

## Des options supplémentaires du paq `scrlayer-scrpage`

À propos des déclarations dans le chapitre 5 de la partie I du présent guide, le paq `scrlayer-scrpage` propose de nombreuses autres possibilités qui sont des extensions que l'utilisateur moyen n'exige pas ou qui existent seulement pour des raisons de compatibilité avec `scrpage2`. Tous ces éléments sont donc documentés ici, dans la Partie II.

### 17.1 Influencer les styles de page définis

Le novice n'utilisera pas nécessairement immédiatement ces éléments.

```
\GenericMarkFormat{nom de classification}
```

Cette déclaration est utilisée par défaut pour mettre en place des numéros d'unité dans l'entête en dessous des sections et des classes sans `\chapter` mais aussi pour le niveau de `section` et de `sous-section`, si les `mark` correspondants ne sont pas déjà définis ailleurs.

L'instruction utilise la valeur par défaut `\@secntmarkformat` si l'instruction interne est définie avec les classes `KOMA-Script`. Sinon, une déclaration avec `\@secntformat` utilise une instruction déjà prévue par `LATEX` pour les classes et les paqs et modifiée quelque peu par `KOMA-Script`.

Comme argument, `\GenericMarkFormat` attend le nom de la structure, par exemple `chapitre` ou `section` sans `controble`<sup>1</sup>. En redéfinissant cette déclaration, le formatage par défaut de tous les numéros d'unité dans les rubriques peut être modifié. En outre, une classe peut spécifier un formatage par défaut différent sans avoir à changer individuellement toutes les commandes.

Si vous voulez que tous les numéros d'unité d'un article dans `Kolumnentitel` affichent le chiffre en blanc sur un fond noir, comme feraient les instructions `\sectionmarkformat` et `\subsectionmarkformat` de *sclayer* en utilisant `\GenericMarkFormat`, il suffit de redéfinir l'instruction appropriée :

```
\documentclass{article}
\usepackage{blindtext}
\usepackage[automark]{sclayer-scrpage}
\pagestyle{scrheadings}
\usepackage{xcolor}
\newcommand*{\numberbox}[1]{\%
\colorbox{black}{\%
\strut~\textcolor{white}{#1}~}\%
}
\renewcommand*{\GenericMarkFormat}[1]{\%
\protect\numberbox{\csname the#1\endcsname}\%
\enskip
}
\begin{document}
\blinddocument
\end{document}
```

Pour les instructions de commutation de couleur, le paq `xcolor` est utilisé. Pour plus de détails, reportez-vous aux instructions sur le paq (voir [Ker07]). En outre, un support invisible est inséré avec `\strut`. Cette déclaration devrait faire partie de toutes les introductions détaillées de L<sup>A</sup>T<sub>E</sub>X.

Pour les cas avec un nombre, la commande d'aide propre à `\numberbox` est définie et protégée dans la redéfinition des `\GenericMarkFormat` avec `\protect` avant l'expansion. Cela est nécessaire parce que sinon avec `\MakeUppercase` pour le réglage en capitales<sup>2</sup> du titre de colonne, ce ne sont plus les couleurs

---

1. Voir la note de la page xxv.

2. Le terme allemand « Versal » (lat.= versus, du côté de la charrue, le sillon de la ligne) est, en typographie, le terme technique pour les lettres « majuscules » ou « supérieures », dérivé du rôle des « capitales » comme un moyen de mettre en évidence

« black » et « white », mais les couleurs « BLACK » et « WHITE » qui sont nécessaires et ne sont pas définies.

Sinon, vous pouvez définir `\numberbox` à l'aide de `\DeclareRobustCommand*` au lieu de `\newcommand*` (voir [Tea06]). Si vous voulez le même livre avec une classe KOMA-Script, ou avec les classes `book` et `report standard`, vous auriez à redéfinir `\sectionmarkformat` et `\subsectionmarkformat` ou en fonction de la classe `\chaptermarkformat` parce que ceux-ci, lorsque vous utilisez les classes mentionnées n'utilisent pas `\GenericMarkFormat` :

```
\documentclass{scrbook}
\usepackage{blindtext}
\usepackage[automark]{sclayer-scrpage}
\pagestyle{scrheadings}
\usepackage{xcolor}
\newcommand*{\numberbox}[1]{\%
\colorbox{black}{\%
\strut~\textcolor{white}{#1}~}\%
}
\renewcommand*{\GenericMarkFormat}[1]{\%
\protect\numberbox{\csname the#1\endcsname}\%
\enskip
}
\renewcommand*{\chaptermarkformat}{\%
\GenericMarkFormat{chapter}\%
}
\renewcommand*{\sectionmarkformat}{\%
\GenericMarkFormat{section}\%
}
\begin{document}
\blinddocument
\end{document}
```

L<sup>A</sup>T<sub>E</sub>X utilisé pour les styles de page est normalement un « deux pièces » de T<sub>E</sub>X. Dans `Kolumnentitel`, on accède au repère de la partie gauche avec `\leftmark` alors que le repère de la partie droite `\rightmark` reste disponible.

---

le type de caractères au début du texte depuis le v<sup>e</sup> siècle. Dans le langage courant, le pluriel est principalement utilisé : « les capitales ». La contrepartie de ces « majuscules » est désignée par les « communes » ou « lettres minuscules ».

<code>\righttopmark</code>
<code>\rightbotmark</code>
<code>\rightfirstmark</code>
<code>\lefttopmark</code>
<code>\leftbotmark</code>
<code>\leftfirstmark</code>

En fait, il est prévu également que `\leftmark` soit utilisé pour les pages de gauche et `\rightmark` pour les pages de droite dans une impression recto-verso. En mode unilatéral, les commandes de sectionnement des classes `standard` considèrent, en priorité, le repère de la partie gauche.

$\TeX$  connaît trois façons d'accéder à un repère.

- `\botmark` sur la dernière compilation de la dernière page de marque valide. Ceci représente la dernière série de repères sur une page. Si aucune marque n'a été mise sur le côté, il correspond à la dernière série de marques qui ont déjà été émises sur les pages. La commande  $\LaTeX$  `\leftmark` utilise précisément ce repère, sur la partie gauche de la note finale de la page. Il en est de même avec `\leftbotmark`. En comparaison, `\rightbotmark` gère les repères de la partie droite.
- `\firstmark`, premier repère de page récemment compilée représente la première marque mise sur le côté et si aucune marque ne figure, correspond à la dernière série de repères déjà émise sur les pages. La commande  $\LaTeX$  `\rightmark` utilise précisément ce repère pour gérer la partie droite de la première marque de la page. Ceci vaut également pour `\rightfirstmark`. En comparaison, `\leftfirstmark` gère les repères de la partie gauche.
- `\topmark` est le contenu qu'avait `\botmark` avant que la page courante ne soit créée.  $\LaTeX$  ne l'a jamais utilisé lui-même mais `scrlayer` offre cette possibilité avec `\lefttopmark` pour un accès à la partie gauche de ce repère et avec `\righttopmark` pour la partie droite.

Il convient de noter que la gauche et la droite du repère ne peuvent être placés ensemble. Même si vous n'avez changé que la partie droite avec `\markright`, la partie gauche (inchangée) est réglée aussi.

En conséquences, dans la mise en page recto-verso, les niveaux de détail les plus élevés des rubriques de la page de style comportent toujours deux pièces.

Utiliser `\chaptermark` puis `\markboth` avec un argument vide à droite. C'est la raison pour laquelle `\rightmark` ou `\firstmark` retourne toujours une valeur vide sur la page d'un titre de chapitre, même si par la suite `\sectionmark` ou indirectement avec `\section`, une nouvelle partie droite du repère est fixée.

Noter que l'utilisation des instructions déclarées ici pour l'édition de la partie gauche ou de la partie droite du repère dans une page peut conduire à des résultats inattendus. Elles sont vraiment destinées à être utilisées dans l'entête ou le pied de page d'un style de page. Par conséquent, elles devraient toujours être une partie du contenu d'un plan de `scrlayer`. En revanche, les confiner à l'arrière-plan ou au premier plan n'a pas d'importance, puisque tous les niveaux ne seront délivrés qu'après la compilation de la page réelle.

Pour plus de détails sur le mécanisme `mark` de `TeX`, se référer à l'exemple dans [Knu90 chapitre 23]. Cette question est traitée ici comme un savoir absolu pour expert. Si les explications ci-dessus vous paraissent plutôt confuses, oubliez-les.

```
\@mkleft{repère gauche}
\@mkright{repère droit}
\@mkdouble{repère}
\@mkboth{repère gauche}{repère droit}
```

Dans les classes et paqs, il arrive que `Kolumnentitel` ne doit pas être réglé seulement lorsque l'exécution automatique est activée (voir l'option `automark` et la commande `\automark`). Avec les classes `standard` cela va exclusivement via `\@mkboth`. Cette déclaration est soit `\@gobbletwo`, déclaration qui détruit leurs deux arguments, ou `\markboth`, instruction qui règle à la fois les repères gauche et droit. Un paq comme `babel` dépendra aussi de `\@mkboth` même, par exemple, pour un changement de langue dans `Kolumnentitel`.

Si vous voulez seulement un seul repère à gauche ou un repère à droite sans modifier l'autre marque, il faut définir des commandes appropriées. Le paq `scrlayer` lui-même exige ce type d'instructions, par exemple, sous les titres automatiques de fonctionnement. `\@mkleft` pour régler seulement la marque gauche, `\@mkright` pour mettre juste la bonne marque ou `\@mkdouble` pour, à la fois, régler les repères droit et gauche avec le même contenu non défini lors du chargement par `scrlayer`, mais définis par le paq lui-même. La définition sélectionnée reconnaît l'état de `\@mkboth` avec des titres automatiques et dans ce cas (et seulement dans ce cas), utilise les commandes correspondantes à un repère.

Classes et paqs d'auteurs peuvent également avoir recours à ces instructions si vous souhaitez ne définir que les repères gauche ou droit lorsque vous travaillez avec des titres automatiques.

## 17.2 Définir vos propres paires de style de page

Dans la section 5.4, les deux styles de page présentés sont `scrheadings` et `plain.scrheadings`. Cet ensemble forme pratiquement une paire dans laquelle `scrheadings` fournit le titre principal du style de page avec les titres courants tandis que `plain.scrheadings` fournit une simple page de style correspondant à la page en cours d'exécution, sans titre, mais généralement avec la pagination, à savoir les numéros de page. En plus de la configuration de cette paire fournie prédéfinie par `scrlayer-scrpage`, il est également possible de définir des paires supplémentaires. Le nom des principaux styles de page de `scrheadings` sert pratiquement de nom de style de la paire de page.

La grande majorité des utilisateurs obtiendra généralement une paire de style de page avec les prédéfinis de `scrheadings`. Les instructions figurant dans cette section sont donc plutôt des suppléments pour les cas particuliers. Il n'y a aucun exemple d'application détaillé dans l'immédiat. Je reviendrai sur ces cas dans les futures versions. Je peux, dans le même temps, faire en sorte que cet ensemble puisse être réglé avec une paire `scrheadings`.

```
\defpairofpagestyles[paire parents]{nom}{définition}  
\newpairofpagestyles[paire parents]{nom}{définition}  
\renewpairofpagestyles[paire parents]{nom}{définition}  
\providepairofpagestyles[paire parents]{nom}{définition}
```

Avec ces déclarations, les paires de styles de page comparées seront définies avec `scrheadings` et `plain.scrheadings`. Ceci est le nom de la page principale de style comparable pour `scrheadings`, qui est conçue pour une utilisation avec l'entête en cours d'exécution. Le nom du style de `plain` correspondant est automatiquement précédé par `plain`. Ce nom est donc aussi le nom du couple et de la page principale de style de cette paire, tandis que `plain.name` est le nom de la page style simple de cette paire.

Si l'argument optionnel de la paire `parents` est indiqué, ceci sera le nom d'une paire de style de page, avec ses paramètres et le nouveau couple sera initialisé. Ainsi, le nouveau couple va pratiquement hériter de la configuration de la paire `parent`.

Alors que dans la section 5.4 l'impression est donnée que les instructions ne s'appliquent qu'aux deux styles de page `scrheadings` et `plain.scrheadings` aussi longtemps qu'ils sont la seule paire de style de pages, si plusieurs styles sont définis, les instructions ne concernent que les deux derniers styles actifs. Cela est vrai pour `\lehead`, `\cehead`, `\rehead`, `\lohead`, `\cohead`, `\rohead`, `\lefoot`,



`\cefoot`, `\refoot`, `\lofoot`, `\cofoot`, `\rofoot`, `\ihead`, `\chead`, `\ohead` ainsi que `\ifoot` `\`, `\ofoot` de la section 5.4.

En plus des dix-huit qui précèdent, trois commandes sont documentées ci-dessous `\clearmainofpairofpagestyles`, `\clearpairofpagestyles` ainsi que `\clearplainofpairofpagestyles` utilisées dans la définition de paramètres. Dans ce cas, elles représentent une sorte de configuration de base de la paire de style de page qui est toujours exécutée lorsque la paire de style de page est activée. Un couple de style de page est activé par l'un des deux styles de page de la paire activée. Cela se fait habituellement avec l'aide de `\pagestyle`.

Il convient de noter que les instructions de la section 5.5 sont, dans tous les cas, de nature générale et s'appliquent à tous les styles de page définis avec `scrlayer-scrpage`.

Alors que `\defpairofpagestyles` définit la paire de style indépendamment du fait que les styles de page correspondant existent déjà, utiliser si les styles de page ne sont pas encore définies :

`\newpairofpagestyles` et `\providepairofpagestyles`.

Si au moins l'un des deux styles de page de la paire est défini, la redéfinition dans `\providepairofpagestyles` est ignorée et `\newpairofpagestyles` provoque une erreur. Pour la redéfinition des paires existantes, `\renewpairofpagestyles` peut être utilisé. Une erreur est signalée, si l'un des styles ou les deux styles de page du couple n'existent pas encore.

<code>\clearmainofpairofpagestyles</code> <code>\clearplainofpairofpagestyles</code> <code>\clearpairofpagestyles</code>
--

Avec `\clearmainofpairofpagestyles`, le style de la page principale active est configuré vide. En revanche, la page style simple de la paire de style de page pertinente est configurée, avec la déclaration `\clearplainofpairofpagestyles`, vide. La commande `\clearpairofpagestyles` configure enfin les deux styles de page de la paire correspondante comme vide.

Aucune des déclarations de définition des paramètres spécifiés lorsque vous fixez le style de page paire (voir ci-dessus) n'est éliminée. Lors de la sélection d'une nouvelle page style du couple, ces paramètres sont donc reconduits.

Les instructions peuvent également être utilisées dans la définition d'une paire de style de page indiquée ci-dessus. Mais vous pouvez aussi l'utiliser à tout moment en dehors de cette définition. Dans ce cas, elles se réfèrent à la paire activée la plus récente. Les commandes `\clearscrheadings`, `\clearscrplain`

et `\clearscrheadfoot` sont des Alias pour ces déclarations et n'existent que pour la compatibilité avec `scrpage2`. Vous ne devriez plus les utiliser.

## 17.3 Définir de simples styles de page avec entête et pied de page en trois parties

En plus de la page de style prédéfinie et de la possibilité de définir vos propres paires de style de page, `scrlayer-scrpage` permet également de définir des styles de page, où entête et pied de page sont partagés en trois éléments. Cette interface existe uniquement pour la compatibilité avec `scrpage2` et un ensemble de commandes plus cohérentes a même été étendu à cette fin. Il est recommandé de ne pas recourir à cette possibilité pour de nouveaux documents mais de faire usage des options clairement supérieures des paragraphes précédents ou de la `section` suivante. Avec ces instructions, un style unique peut être défini, dans l'entête et le pied de page, divisés en trois éléments : un élément interne, un élément intermédiaire et un élément extérieur. Ils sont donnés par trois arguments obligatoires.

L'élément interne se trouve sur le côté gauche, aligné à droite et placé dans l'entête et le pied de page. Placé sur le côté droit, il est justifié à gauche. En conséquence, l'élément intermédiaire est centré sur les deux côtés gauche et droit de l'entête ou du pied de page. L'élément extérieur est finalement aligné à gauche sur le côté gauche et à droite sur le côté droit de l'entête et du pied de page. La page de droite n'existe que dans un côté verso.

Ces deux arguments optionnels permettent d'activer des lignes dans l'entête et le pied de page en ajoutant l'épaisseur souhaitée comme argument de ligne.

- Le premier argument se rapporte à l'épaisseur de la ligne extérieure. Cette ligne est tracée sur l'entête et sous le pied. L'argument est facultatif.
- Le second argument optionnel concerne l'épaisseur des lignes intérieures. Ces deux lignes sont tracées sous l'entête et au-dessus du pied.
- Si un seul argument optionnel est donné, ce sera l'épaisseur des lignes intérieures. L'épaisseur des lignes extérieures est égale à zéro, ne laissant dessinées que les deux lignes intérieures.
- Si aucun des arguments optionnels n'est spécifié alors, toutes les lignes sont omises.

### 17.3 Définir de simples styles de page avec entête et pied de page en trois parties

<code>\deftriplepagestyle</code>	$\{nom\ du\ style\ de\ page\}$ <i>[épaisseur de ligne extérieure]</i> <i>[épaisseur de ligne intérieure]</i> $\{élément\ d'entête\ intérieure\}$ $\{élément\ d'entête\ moyenne\}$ $\{élément\ d'entête\ externe\}$ $\{élément\ de\ pied\ intérieur\}$ $\{élément\ de\ pied\ moyen\}$ $\{élément\ de\ pied\ extérieur\}$
<code>\newtriplepagestyle</code>	<i>nom du style de page</i> <i>[épaisseur de ligne extérieure]</i> <i>[épaisseur de ligne intérieure]</i> $\{élément\ d'entête\ intérieure\}$ $\{élément\ d'entête\ moyenne\}$ $\{élément\ d'entête\ externe\}$ $\{élément\ de\ pied\ intérieur\}$ $\{élément\ de\ pied\ moyen\}$ $\{élément\ de\ pied\ extérieur\}$
<code>\renewtriplepagestyle</code>	$\{nom\ du\ style\ de\ page\}$ <i>[épaisseur de ligne extérieure]</i> <i>[épaisseur de ligne intérieure]</i> $\{élément\ d'entête\ intérieure\}$ $\{élément\ d'entête\ moyenne\}$ $\{élément\ d'entête\ externe\}$ $\{élément\ de\ pied\ intérieur\}$ $\{élément\ de\ pied\ moyen\}$ $\{élément\ de\ pied\ extérieur\}$
<code>\providetriplespagestyle</code>	$\{nom\ du\ style\ de\ page\}$ <i>[épaisseur de ligne extérieure]</i> <i>[épaisseur de ligne intérieure]</i> $\{élément\ d'entête\ intérieure\}$ $\{élément\ d'entête\ moyenne\}$ $\{élément\ d'entête\ externe\}$ $\{élément\ de\ pied\ intérieur\}$ $\{élément\ de\ pied\ moyen\}$ $\{élément\ de\ pied\ extérieur\}$

Noter que les arguments optionnels vides ne sont pas équivalents à l’omission des arguments optionnels. Les arguments optionnels vides ne sont pas autorisés dans ce cas. En revanche, les conditions de réglage de simple longueur de la largeur de l’entête permettent déjà un choix (voir [section 5.5](#)).

La signification de chaque argument est illustrée dans un schéma des deux pages dans la figure ci-dessous. Voici des exemples de contenus « extérieur », « centre » et « intérieur ». Les flèches associées illustrent la propagation de ces contenus. Le texte désigne les noms des arguments des commandes. Les flèches grisées indiquent les éléments correspondants de la page.

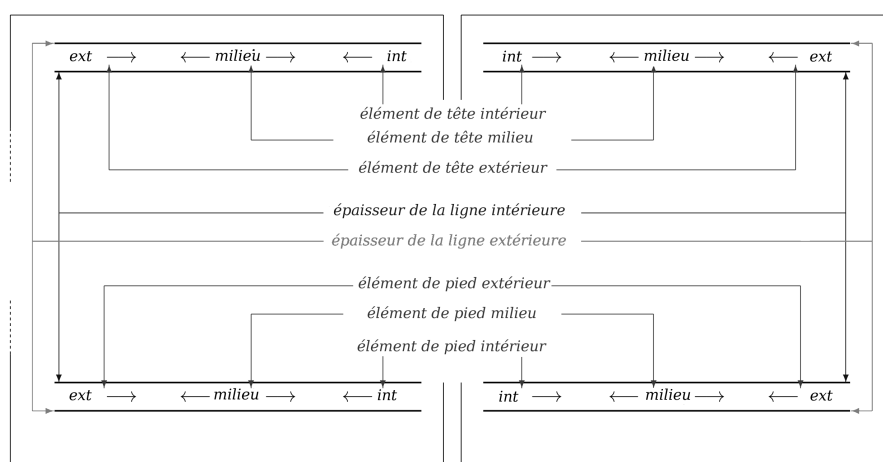


FIGURE 17.1 – Éléments des trois parties d’un style de page double face avec les instructions `deftriplepagestyle`, `newtriplepagestyle`, `providetripelapagestyle` et `renewtriplepagestyle`

Avec les instructions décrites, il est impossible d’activer les lignes individuelles, par exemple, une seule ligne en dessous de l’entête de la page, mais pas de ligne au-dessus du pied de page, les utiliser à la place de l’une des instructions précédentes ou de la [section](#) suivante.

La longueur des éléments individuels n’est pas limitée par la présence d’un élément voisin. Il peut se produire que, en cas d’utilisation défavorable, un élément écrase un ou deux éléments voisins. L’utilisateur est responsable pour se soustraire à ce cas défavorable. Lors de l’exécution du titre, cela est possible, par exemple, en utilisant l’argument optionnel de commandes de sectionnement. Voir les instructions pour les commandes de division des classes respectives.

Alors que `\deftriplepagestyle` définit un style de page, indépendamment du fait qu'il a déjà été défini précédemment, `\newtriplepagestyle` ainsi que `\providetripstyle` définissent seulement les styles de page qui ne sont pas encore définis.

Si un style de page, déjà défini, est obtenu avec `\newtriplepagestyle`, alors `\providetripstyle` provoque un message d'erreur et le rejet de cette nouvelle définition. `\renewtriplepagestyle` permet de redéfinir un style de page existant. Cette déclaration signale une erreur si un style avec ce nom n'est pas encore défini. La commande `\deftripstyle` du paq `scrpage2`, correspond à la nouvelle commande `\deftriplepagestyle`. Rebaptisée pour unifier les noms des instructions, elle ne devrait plus être utilisée. Si elle l'est encore, cela se traduit par un avertissement.

## 17.4 Définition des styles de page complexes

En plus de la page de styles prédéfinis et de la création de vos propres styles de page, `scrlayer-scrpage` fournit une manière simple de définir ces derniers. Les concepts décrits dans les sections précédentes en permettent la mise en œuvre en revenant sur cette possibilité. En raison de sa complexité, l'usage de cette interface est uniquement recommandé aux utilisateurs expérimentés qui peuvent, en outre, ignorer la suite de ce chapitre. Si ce n'est pas votre cas, vous pouvez déjà réaliser, avec les pré-citées, presque tout ce qui est possible avec cette interface de base.

```
\defpagestyle{nom}{définition d'entête}{définition de pied de page}
\newpagestyle{nom}{définition d'entête}{définition de pied de page}
\providepagestyle{nom}{définition d'entête}{définition de pied de page}
\renewpagestyle{nom}{définition d'entête}{définition de pied de page}
```

Ces instructions destinées à la définition d'un seul style avec un maximum de flexibilité s'appliquent au nom du style de page à spécifier.

Les deux paramètres caractérisant l'entête et le pied de page ont une structure identique :

```
(longueur ligne supérieure, épaisseur ligne supérieure)%
(définition des pages gauche en disposition recto-verso)%
(définition des pages droite en disposition recto-verso)%
```

(définition des pages droite en disposition unilatérale)%  
(longueur ligne inférieure, épaisseur ligne inférieure)

Les arguments entre parenthèses sont facultatifs, ce qui signifie qu'ils peuvent être omis, ainsi que les parenthèses. Dans ce cas, les longueurs et épaisseurs des lignes seront réglées selon les options `headtopline`, `headsepline`, `footsepline` et `footbotline` (voir section 5.5).

TABLE 17.1 – Noms des niveaux du style de page définis par `sclayer-scrpage`

Nom du niveau	Signification du niveau
<code>nom.head.above.line</code>	la ligne au-dessus de l'entête
<code>nom.head.odd</code>	côté droit de l'entête mise en page recto-verso
<code>nom.head.even</code>	côté gauche de l'entête mise en page recto-verso
<code>nom.head.oneside</code>	l'entête de page mise en page unilatérale
<code>nom.head.below.line</code>	la ligne sous l'entête
<code>nom.foot.above.line</code>	la ligne au-dessus du pied de page
<code>nom.foot.odd</code>	côté droit du pied de page mis en page recto-verso
<code>nom.foot.even</code>	côté gauche du pied de page mis en page recto-verso
<code>nom.foot.oneside</code>	le pied de page mis en page unilatérale
<code>nom.foot.below.line</code>	la ligne sous le pied de page

Les trois arguments avec les définitions sont obligatoires et sont utilisés en fonction de la mise en page et des paramètres. Avec `Kolumnentitel`, le contenu de la définition est arbitraire, mais l'utilisation de `\headmark`, de `\leftmark` ou de `\rightmark` est recommandée dans les définitions de style de page. En aucun cas, vous ne devez spécifier directement un numéro de classification ou un texte comme titre dans `Kolumnentitel`. En raison de la structure de page asynchrone de `LATEX`, un mauvais numéro ou un texte incorrect peuvent apparaître ailleurs que dans l'entête de page.

Dans `\defpagestyle` le style de page est redéfini, existant déjà ou non. En revanche, `\newpagestyle` émet une erreur quand un style de page avec le même nom existe déjà et la définition de `\providepagestyle` est ignorée si le nom a déjà été utilisé pour un style de page. Inversement, `\renewpagestyle` permet de redéfinir un style de page existant mais une erreur s'affiche en cas de changement de nom de cette déclaration.

Les quatre déclarations sont basées sur `\DeclarePageStyleByLayers` du paquet `scrlayer`. Les niveaux définis ici pour le nom du style de page sont présentés dans la table 17.1 page ci-contre. Pour plus d'informations sur ces niveaux et les niveaux des styles de page, se référer au chapitre précédent.

Si, par exemple, vous voulez utiliser une couleur dans l'entête entière du style de page `scrheadings`, grâce à l'introduction de ce chapitre, vous savez que `scrheadings` est un style de page utilisant `scrheadings.head.onside`, `scrheadings.head.even` et `scrheadings.head.odd`. Vous pouvez définir trois niveaux pour le fond et les coller au début du style de page :

```
\documentclass{scrartcl}
\usepackage[automark]{scrlayer-scrpage}
\usepackage{xcolor}
\usepackage{blindtext}
\DeclareLayer[clone=scrheadings.head.onside,
contents={\%
\color{yellow}\%
\rule[-\dp\strutbox]\%
{\layerwidth}{\layerheight}\%
}\%
]{scrheadings.head.onside.background}
\DeclareLayer[clone=scrheadings.head.odd,
contents={\%
\color{yellow}\%
\rule[-\dp\strutbox]\%
{\layerwidth}{\layerheight}\%
}\%
]{scrheadings.head.odd.background}
\DeclareLayer[clone=scrheadings.head.even,
contents={\%
\color{yellow}\%
\rule[-\dp\strutbox]\%
{\layerwidth}{\layerheight}\%
}\%
]{scrheadings.head.even.background}
\AddLayersAtBeginOfPageStyle{scrheadings}{\%
scrheadings.head.onside.background,\%
scrheadings.head.odd.background,\%
scrheadings.head.even.background\%
```

```
}
\pagestyle{scrheadings}
\begin{document}
\blinddocument
\end{document}
```

Comme vous pouvez le voir, trois niveaux ont été utilisés dans l'exemple, de sorte que, pour la position et la taille des niveaux de fond, il suffit simplement de copier, grâce à l'option `clone`, chacun des niveaux de l'entête, ce qui est plus facile que de déterminer la position d'une couche de fond.

Le fond de couleur, lui-même, est défini dans cet exemple avec une commande `\rule`. Quant à la taille, `\layerwidth` est utilisé pour la hauteur de la couche et `\layerheight` pour la largeur. Ce peut être décalé vers le bas par l'intermédiaire d'un argument optionnel `\dp\strutbox`.

Au lieu d'ajouter de nouveaux niveaux comme exemple de la couleur de fond, nous sommes en mesure de résoudre ce problème avec `\` et `\chead`. Il est toujours recommandé de travailler à une solution appropriée comme dans un exercice. De même, il faudrait des niveaux de fond individuels à insérer chaque fois immédiatement avant le contenu du niveau pertinent, ce qui est aussi recommandé comme un exercice.

`hmode=off-on-valeur`

Dans `scrpage2`, l'entête et le pied de page étaient généralement émis en mode horizontal. Dans `scrlayer-scrpage`, le lien est en mode horizontal par défaut. Si l'option `hmode` est activée, `scrlayer-scrpage` se comportera d'une manière compatible avec `scrpage2` avant la sortie commutée en mode horizontal, ce qui peut affecter, lors de la première édition, à la fois le traitement de l'espace ainsi que l'orientation verticale.

Cette option peut prendre les valeurs standard pour de simples interrupteurs qui sont présentées dans la table du chapitre « Recalculer avec `typearea.sty` ». Par défaut, le commutateur est désactivé.



# Chapitre 18

## Notice sur sclayer-notecolumn

Jusqu'à, et y compris, la version `KOMA-Script` 3.11b, les notes<sup>1</sup> peuvent être fournies avec un contenu `\marginpar` et `\marginline` (voir la section 3.21).

Cependant, ce genre de notes en marge présente quelques inconvénients :

- les notes de marge ne peuvent être pleinement mis en œuvre sur une seule page,
- les sauts de page dans les notes marginales ne sont pas possibles, ceci résulte en partie du fait qu'elles dépassent du bord inférieur, vers le bas ;
- les notes marginales près du saut de page peuvent glisser sur la page suivante et sont émises dans le cas où la mise en page recto-verso avec une alternance marginale de colonnes serait incluse dans le mauvais bord. Ce problème est résolu avec le paq supplémentaire `mparhack` ou en utilisant `\marginnote` du paq `marginnote` ;
- une note de marge n'étant pas possible dans un environnement flottant, ce problème est résolu avec `marginnote` ;

---

1. PAPIRAY précise : cet article est, en partie, l'adaptation du chapitre du manuel allemand de `KOMA-Script` complétée, en 18.5, par un extrait du manuel anglais de mise en œuvre provisoire du paq primitif `sclayer-notecolumn` sous la forme de colonnes marginales qui peuvent être fournies avec le contenu. Tant que ce dernier reste primaire, une démonstration du concept n'a que peu d'importance. Malgré cela, une traduction de l'allemand vers l'anglais (et pourquoi pas vers le français...) sera toujours la bienvenue.

- pour des documents `recto-verso`, une seule colonne marginale lorsque vous travaillez avec `reversemarginpar` et `normalmarginpar`, deux tout au plus, et dans lesquels `reversemarginpar` est à peine utilisable.

Mais attention, l'utilisation de `\marginnote` pose un autre problème : Comme le paq n'a pas de détection de collision, de nombreuses notes, qui ont été lancées à proximité, peuvent mutuellement se chevaucher en partie ou en totalité.

En outre, `\marginnote`, en fonction des paramètres sélectionnés, provoque parfois des changements dans l'espacement des lignes du texte courant.

Le paq `sclayer-notecolumn` est un premier pas pour résoudre tous ces problèmes. À cet effet, il repose sur la fonctionnalité de base de `sclayer`, mais ce peut être un désagrément car les notes ne seront exploitées que sur des pages développées avec le style de `sclayer`.

Cet inconvénient peut être facilement résolu en utilisant `sclayer-scrpage`, et peut même se transformer en un avantage.

## 18.1 Information pour le développement

Le paq prévu d'abord pour démontrer le potentiel de `sclayer`, conçu comme un soi-disant concept et bien qu'en cours actuel de développement, offre avec `sclayer-notecolumn` et `sclayer` une stabilité de bon aloi. Pourtant, on peut supposer encore des erreurs dans `sclayer-notecolumn` et il est recommandé d'en signaler l'emplacement. Certaines lacunes du paq doivent être réduites et les dépenses d'énergie optimisées. Il est bon de noter que les colonnes peuvent, par exemple, être réparties sur plusieurs pages, mais qu'aucun changement de paragraphe n'a fait l'objet d'une mise en page de remplacement, simplement parce que ce n'est pas possible avec `TEX`. Comme le paq est encore considéré actuellement comme expérimental, les instructions de cette deuxième partie de `KOMA-Script` sont destinées principalement aux utilisateurs avertis.

Pour les débutants, les utilisateurs sur le chemin de `LATEX` et les experts, certaines des explications suivantes peuvent paraître incompréhensibles ou pour le moins ambiguës. Je demande votre indulgence mais je préfère dépenser mon énergie dans des paqs expérimentaux plutôt que dans un manuel d'instructions à demi supportables.

## 18.2 Choix de l'option de sélection précoce ou tardive

Ce qui a été écrit dans la section 2.4 s'applique *mutatis mutandis*.

## 18.3 Textes et marques

Ce qui est écrit dans la section 3.6 est applicable *mutatis mutandis*. Donc, si vous avez déjà lu et compris la section 3.6, vous pouvez passer directement à la section 3.10.

L<sup>A</sup>T<sub>E</sub>X comprend un ensemble d'instructions pour le balisage, en plus du choix de polices pour inclure des commandes, sélectionner la taille ou l'alignement du texte. Plus de détails sur les possibilités normalement définies en consultant [DGS+12], [Tea05a] et [Tea05b].

```
\setkomafont{élément}{commande}
\addtokomafont{élément}{commande}
\usekomafont{élément}
```

Dans le noyau L<sup>A</sup>T<sub>E</sub>X la déclaration `\textsuperscript` déjà définie peut être mise en avant avec `texte`. Une version correspondante pour placer le texte en indice n'est fournie que depuis la version 2015.01.01 de L<sup>A</sup>T<sub>E</sub>X. Pour les anciennes versions, KOMA-Script définit donc `\textsubscript`. Un exemple d'application peut être trouvé dans la section 3.6.

En suivant les instructions `\setkomafont` et `\addtokomafont`, il est possible de définir les commandes modifiant les caractéristiques d'un élément commuté particulier. Théoriquement toutes sortes de déclarations sont utilisées comme commandes, y compris du texte littéral. Cependant, vous devez vous limiter à ces déclarations avec un seul attribut de police activé.

Habituellement, ce sont les commandes `\rmfamily`, `\sffamily`, `\ttfamily`, `\mdseries`, `\bfseries`, `\upshape`, `\itshape`, `\slshape`, `\scshape` ou l'une des commandes `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize`, `\small`, `\footnotesize`, `\scriptsize` et `\tiny`. Pour le détail de ces commandes, consulter [DGS12], [Tea05a] ou [Tea05b]. Les commutations de couleur telle que `\normalcolor` sont possibles (voir [Car05] et [Ker07]). Des instructions permettant les redéfinitions ou gérant la sortie ne sont pas fournies. Dans ce cas un comportement étrange, qui ne constitue pas un bug, est possible.

Un élément de commutation de la police `\setkomafont` est fourni avec une toute nouvelle définition. En revanche, la définition existante est prolongée avec `\addtokomafont`. Il n'est pas recommandé d'utiliser les deux déclarations dans le même document, mais uniquement dans le préambule. Pour des exemples d'utilisation, se référer aux sections sur les différents éléments. Les noms et les significations des éléments individuels sont énumérés dans la table 3.2 page 176 et les paramètres par défaut décrits dans les sections respectives.

Avec la déclaration `\usekomafont` la police en cours peut être changée pour celle de l'élément spécifié. Voir la section 3.6 pour un exemple de l'utilisation de `\setkomafont` et de `\usekomafont`.

```
\usefontofkomafont{élément}
\useencodingofkomafont{élément}
\usesizeofkomafont{élément}
\usefamilyofkomafont{élément}
\useseriesofkomafont{élément}
\useshapeofkomafont{élément}
```

Parfois, lors de la mise en écriture d'un élément, les choses sont ainsi faites qu'elles n'ont rien à voir avec cette écriture, bien que nettement recommandées. Vous devez alors régler uniquement la police et donc utiliser `\usekomafont` et `\usefontofkomafont`, mais aucun des paramètres supplémentaires ne seront effectués. Cette déclaration n'accepte que la taille de la police et de la ligne de base, le codage (eng. *encoding*), la famille (eng. *family*), l'épaisseur de la ligne et l'alignement, la police standard (eng. *fontseries*) et la forme de police (eng. *fontshape*).

Les attributs individuels de police peuvent être adoptés avec les instructions restantes. Ici `\usesizeofkomafont` accepte à la fois les tailles de la police et de la distance de base.

Ces commandes ne doivent pas être interprétées comme une légitimation pour toutes les instructions d'insertion de paramètres d'éléments de police. Cela peut très rapidement conduire à des erreurs (voir la section 21.3)

## 18.4 Déclarer de nouvelles colonnes de notes

Au chargement du paq, une colonne de note est automatiquement déclarée et intitulée `marginpar`. Comme son nom l'indique, elle se trouve sur la colonne marginale à partir de `\marginpar` et `\marginline`. Il faut également noter

`\reversemarginpar` et `\normalmarginpar`, non pour les notes individuelles, mais pour les notes entières d'un côté. Le principal facteur correspond aux paramètres appliqués en fin de page, pour l'édition de la colonne de note. Ceux-ci sont, par défaut, les paramètres pour toutes les nouvelles colonnes déclarées correspondant aux conditions mentionnées pour une mise en page prédéfinie `\marginpar`, ce qui peut facilement être modifié dans la déclaration. Si vous voulez, dans une page, les utiliser à la fois comme notes mais aussi à côté du texte principal, vous devez créer une deuxième colonne de note. Il est important de noter que les mémo colonnes ne sont délivrés qu'aux pages dont le style de page est basé sur le paq `scrlayer` qui chargera automatiquement la colonne `scrlayer-note`, par défaut le style de page vide. En outre, si les styles de page supplémentaires le nécessitent, le paq `scrlayer-scrpage` est recommandé. Ces instructions permettent de créer les colonnes de note.

La commande `\DeclareNoteColumn` crée la colonne de note indépendamment du fait qu'elle existe ou pas, tandis que `\DeclareNewNoteColumn` renvoie une erreur si le nom de la colonne de la note est déjà affecté à une autre colonne de mémo et la commande `\ProvideNoteColumn`, dans ce cas, très simplement, ne fait rien. Avec `\RedeclareNoteColumn`, une seule colonne existante peut être reconfigurée.

Lors de la restructuration de colonnes existantes avec `\DeclareNoteColumn` ou `\RedeclareNoteColumn`, les notes présentes dans ces colonnes ne sont pas perdues, mais conservées. Pour les nouvelles colonnes de note, il est toujours possible de modifier les attributs de police, s'ils n'existent pas encore, à l'aide de `\setkomafont` et `\addtokomafont`.

Chaque colonne de note est référencée par son nom, défini par l'argument « notecolumn.name » obligatoire. En conséquence, il existe pour la colonne prédéfinie `marginpar` l'élément `notecolumn.marginpar`. La valeur par défaut peut s'appliquer à la déclaration d'une police de la colonne mémo directement sur l'option dans la liste des paramètres facultatifs qui peuvent être précisés. La liste des paramètres est une liste d'options séparées par des virgules. Les options disponibles sont données dans le tableau suivant. Par défaut `marginpar` est toujours défini, mais peut être remplacé par des paramètres individuels. Etant donné que les colonnes de notes sont définies à l'aide de `scrlayer`, un niveau est créé pour chaque colonne de la note. Le nom de ce niveau est utilisé également comme nom de la colonne de note. Voir le paragraphe 16,4 pour plus de détails sur les niveaux.

TABLE 18.1 – Réglages possibles pour la déclaration de colonnes de notes

---

Éléments
<hr/> <code>\font=attributs de police</code> <hr/>
définit les attributs de police de la colonne de note en utilisant <code>\setkomafont</code> . La section 3.6, fait référence aux valeurs autorisées. Par défaut : vide
<hr/> <code>\marginpar</code> <hr/>
la position ainsi que la largeur de la colonne de notes sont ajustées afin de correspondre à la colonne marginale <code>\marginpar</code> . La commutation entre <code>\reversemarginpar</code> et <code>\normalmarginpar</code> est observée en sortie de colonne de note ainsi qu'à la fin de la page. Pas d'argument pour cette option. Par défaut : oui.
<hr/> <code>\normalmarginpar</code> <hr/>
la position et la largeur de la colonne sont ajustées afin de correspondre à la colonne marginale <code>\marginpar</code> dans la mise en forme avec <code>\normalmarginpar</code> . Il convient de noter que cette option n'attend ou ne permet aucun argument. Par défaut : non.
<hr/> <code>\position=distance</code> <hr/>
La colonne de note est fixée à une distance du bord gauche du papier. Même des expressions complexes sont autorisées pour la distance, tant qu'elles sont entièrement extensibles et se développent au moment de la production de la colonne à une taille, une valeur de longueur ou une longueur d'impression. Voir [Tea98], section 3.5] pour plus d'informations sur les termes de longueur. Par défaut : avec l'option <code>marginpar</code> .
<hr/> <code>\width=largeur</code> <hr/>
La colonne de note est réglée sur la largeur spécifiée. Même les expressions complexes sont autorisées pour la largeur, si elles sont entièrement extensibles et se développent au moment de la production de la colonne à une taille, une valeur de longueur ou une longueur d'impression. Voir [Tea98], section 3.5] pour plus d'informations sur les termes de longueur. Avec l'option <code>marginpar</code> par défaut.
<hr/> <code>\reversemarginpar</code> <hr/>
La position et la largeur de la colonne sont ajustées pour correspondre dans la mise en forme avec <code>\reversemarginpar</code> à la marginalisation de <code>\marginpar</code> . Il convient de noter que cette option n'attend ou ne permet aucun argument. Par défaut : non.

---

Vous êtes un professeur excentrique de droit et vous avez envie d'écrire un traité sur la « Nouvelle loi sur la diffusion exubérante de blagues générales, GüdaVaS ». L'accent principal sera mis sur le commentaire de paragraphes individuels. Vous optez pour une disposition en deux colonnes, le commentaire inclus dans la colonne principale et chaque paragraphe en couleur, juste à côté, dans une colonne étroite de note :

```

\documentclass{scrartcl}
\usepackage[french,german]{babel}
\usepackage{selinput}
\SelectInputMappings{
    adieresis={ä},
    germandbls={ß},
}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage{xcolor}
\usepackage{scrjura}
\setkomafont{contract.Clause}{\bfseries}
\setkeys{contract}{preskip=-\dp\strutbox}
\usepackage{scrlayer-scrpage}
\usepackage{scrlayer-notecolumn}
\newlength{\paragraphscolwidth}
\AfterCalculatingTypearea{\%
    \setlength{\paragraphscolwidth}{.333\TeXtwidth}\%
    \addtolength{\paragraphscolwidth}{-\marginparsep}\%
}
\recalctypearea
\DeclareNewNoteColumn[\%
position=\oddsidemargin+1in
    +.667\TeXtwidth
    +\marginparsep,
width=\paragraphscolwidth,
font=\raggedright
\footnotesize
    \color{blue}
]{paragraphs}

```

Il présente un article unilatéral. Pour cela, la langue est définie en utilisant le paq de babel en allemand (réformé). L'encodage d'entrée est automatiquement

déterminé en utilisant `\selinput`. La police `Latin Modern` est utilisée dans l'encodage 8 bits. Pour les paramètres de couleur, le paq utilisé est `xcolor`.

Pour un style de page avec le numéro de page, le paq `sclayer-scrpage` est utilisé, ce qui permet d'imprimer les colonnes mémo sur tous les côtés.

Puis le paq `sclayer-notecolumn` est chargé pour les colonnes. La largeur souhaitée de la colonne est recalculée avec `\AfterCalculatingTypearea` après chaque nouveau calcul possible de la zone de type. La distance entre le texte et la colonne de note devrait représenter un tiers de la largeur de la zone pris sur la colonne. Ce que `\marginparsep` permet efficacement. La colonne de notes peut alors être définie avec cette information. Pour déterminer la position, une simple longueur d'impression est utilisée. Il convient de noter que `\oddsidemargin` colle entièrement à tout le bord gauche, mais, pour des raisons historiques, après déduction d'une marge de 1 pouce. Par conséquent, cette valeur doit être prise en compte.

Cette déclaration est terminée. Il est à noter que la colonne de note est affichée jusqu'ici dans la zone de texte. La colonne de note voudrait ainsi remplacer le texte :

```
\begin{document}
\title{Commentaire sur GūdaVaS}
\author{Professeurs R. O. Tenaze et R. Ougenez}
\date{11.\,11.\~2011}
\maketitle
\tableofcontents
\section{Préambule}
```

Le GūdaVaS est probablement la loi la plus importante adoptée en matière de plaisanterie dans les cent mille dernières années. On trouve trace d'une première lecture, le 11\11\11 au congrès au sommet des maniaques de la plaisanterie, mais est-ce vraiment une première qui fût rejetée par l'ancien vizir du divertissement. C'est après la conversion de l'humeur maniaque représentant la monarchie fols que le triomphe du spirituel et irrésistible W. Itzbold (et de son cousin français F. Arceur) que s'ouvre enfin, le 9 septembre 1999, la voie pour cette loi.

Par conséquent, la zone de texte n'étant pas réduite, l'ensemble des suppléments sera imprimé sur toute la largeur. Pour tester l'exemple, vous pouvez ajouter de manière temporaire :

```
\end {document}.
```



Il reste la question de savoir comment le texte pour le commentaire peut être réglé plus étroit, ce qui sera découvert dans la suite de l'exemple.

## 18.5 Créer un mémo

Après la déclaration d'une colonne mémo, les notes peuvent être créées pour cette colonne. Ces notes ne sont pas émises immédiatement, mais d'abord écrites dans un fichier auxiliaire avec l'extension `slnc`. Lors de la prochaine exécution de  $\text{\LaTeX}$ , ce fichier d'aide est lu pièce par pièce, en fonction de l'avancement du nouveau document et distribue les notes de chaque côté à la fin de la page. Il convient de noter que les colonnes mémo sont imprimées uniquement sur les pages dont le style est basé sur le paq `scrlayer` qui charge automatiquement `scrlayer-notecolumn` et fournit par défaut un style de page `empty`. Si des styles de page supplémentaires sont nécessaires, le paq `scrlayer-scrpage` est aussi recommandé.

```
\makenote[nom de la colonne de note]{note} \makenote*[nom de la
colonne de note]{note}
```

Grâce à cette commande, une note peut être créée. La position verticale est utilisée comme position verticale pour le début et la position horizontale découle de la position définie de la colonne de note. Pour un fonctionnement correct, le paq charge les instructions `\pdfsavepos`, `\pdflastypos` et `\pdfpageheight`. Sans ces commandes `scrlayer-notecolumn` ne fonctionne pas. Il est également supposé que les dites primitives produisent des résultats exacts dans  $\text{\PDFTeX}$ .

Si une collision est détectée dans la colonne avec une `note` précédente, la `note` est décalée en dessous de cette `note` antérieure. Une `note` qui n'est plus associée à une page est mise en pages en partie ou en totalité sur la page suivante. La colonne de destination de la `note` créée est déterminée par le nom de l'argument optionnel de la colonne de `note`. Si aucun argument facultatif n'est spécifié, la colonne prédéfinie `marginpar` est utilisée.

Ajoutons maintenant, à l'exemple précédent, un paragraphe commenté où la `section` est destinée à être placée dans la même colonne que la `note` redéfinie :

```
\section{Analyse}
\begin{addmargin}[Opt]{.333\TeXtwidth}
  \makenote[paragraphs]{\%
  \protect\begin{contract}
```

```
\protect\Clause{\%
  {title={Pas de blague sans public}}\%
}
  Une plaisanterie ne peut être drôle
que si elle rencontre une foule . \protect
\protect\end{contract}\%
}
```

Voici la règle de base tellement élémentaire qu'il est tout à fait fondé de s'incliner devant la sagesse de son auteur.

Dans un premier temps, les notes sont écrites dans un fichier `.aux`, puis après lecture de ce fichier `\end{document}`, transférées dans un fichier `.snlc`. Le réglage `\nofiles` est ignoré, le cas échéant. L'environnement documenté dans l'article 3.17, est utilisé pour réduire les paragraphes du texte principal à la largeur de la colonne. C'est aussi l'un des rares problèmes rencontrés en utilisant `\makenote`.

Les commandes peuvent être réparties, malheureusement, dans l'argument depuis que les paramètres obligatoires sont écrits dans les fichiers. Inclure `\protect` avant toutes les commandes permet de l'éviter sinon l'utilisation des instructions au sein de cet argument peut conduire à un message d'erreur.

En principe, vous pouvez maintenant clore l'exemple avec :

```
\end{addmargin}
\end{document}
```

si vous voulez en découvrir le résultat.

Lors du test de l'exemple, vous verrez que la colonne des lois descend plus profond que le texte du commentaire. Lors d'une application exécutée dans le but d'ajouter un paragraphe à une section, le problème du commentaire qui ne continue pas au-dessous du texte juridique, mais directement après le commentaire précédent peut éventuellement se poser. Un problème que vous apprendrez à résoudre.

<code>\syncwithnotecolumn[nom de la colonne de note]</code>
---

Un point de synchronisation est créé dans une colonne de mémo, grâce à cette commande, mais jamais dans le texte principal du document. Quand un tel point de synchronisation est atteint lors de l'émission d'une note ou d'une colonne de texte principal, une marque est créée dont le contenu est la page actuelle et la position verticale de ce point dans cette page.

Parallèlement à la création du point de synchronisation, il est vérifié si une marque existe dans le texte principal de la colonne de note depuis la dernière exécution de  $\LaTeX$ . Si tel est le cas, leurs valeurs sont comparées. Si la marque se trouve en bas de page ou sur la page suivante, la colonne est avancée jusqu'au point dans le texte principal.

Généralement, la synchronisation des points ne devrait pas être réalisée dans un paragraphe du texte principal, mais seulement entre eux.

Si `\syncwithnotecolumn` est néanmoins utilisé dans un paragraphe, le point de synchronisation dans le texte principal est effectivement inséré après la ligne actuelle. À cet égard, `\syncwithnotecolumn` ressemble par exemple à `\vspace`.

Le mécanisme qui fait que les points de synchronisation sont détectés dans la colonne de note jusqu'à la prochaine exécution de  $\LaTeX$  nécessite au moins trois points de  $\LaTeX$ . Chaque nouvelle synchronisation peut en outre entraîner des changements pour les points de synchronisation ultérieurs, et à son tour nécessiter une nouvelle exécution de  $\LaTeX$  lui-même. Ces changements sont susceptibles de provoquer le message suivant : «  $\LaTeX$  Warning : Label(s) may have changed » ( $\LaTeX$  attention : l'étiquette peut avoir changé). Recompiler pour obtenir des références croisées à droite. Mais des messages sur les étiquettes non définies peuvent aussi tout simplement attirer l'attention sur la nécessité de poursuivre le fonctionnement de  $\LaTeX$ .

Si l'argument optionnel est omis, `marginpar` le remplace, afin d'utiliser la colonne de note prédéfinie.

Dans une note, `\syncwithnotecolumn` n'est pas autorisé, donc il faut utiliser l'argument obligatoire `\makenote`. Cette erreur ne peut pas, actuellement, être supprimée et conduit à des déplacements qui se produisent à chaque nouvelle exécution de  $\LaTeX$  et empêchent d'atteindre l'état final. Pour coordonner deux ou plusieurs colonnes ensemble, il est préférable de les mettre en concordance avec le texte principal, qui, à son tour, les synchronisera les unes avec les autres. L'instruction préconisée à cet effet est décrite ci-dessous.

Nous illustrons maintenant la création d'un point de synchronisation puis d'un paragraphe pour ajouter un commentaire :

```
\syncwithnotecolumn[paragraphs]\bigskip
\makenote[paragraphs]{\%
\protect\begin{contrat}
\protect\Paragraph{title={Comédie de la culture}}
\setcounter{par}{0}\%
```

```
Le comique d'une blague est déterminé, par
l'environnement culturel dans laquelle elle est dite,
Le comique d'une blague est déterminé, par
l'environnement culturel dans laquelle elle influe.
\protect\end{contract}
}
```

Bien sûr, le volet culturel de la plaisanterie n'est pas négligeable, même si la rectitude politique dépendante de l'utilisation de cet environnement culturel peut être combattue admirablement, l'impact de l'humour développé dans un cadre approprié reste frappant. De la même manière, on peut supposer que dans un mauvais environnement culturel une blague peut devenir également un réel danger pour le conteur de plaisanterie.

En plus du point de synchronisation, une distance verticale est aussi insérée avec `\bigskip` pour séparer les paragraphes individuels des commentaires.

Aussi, voici un autre point qui peut être un problème visible. Depuis que la colonne de note travaille avec des boîtes montées et démontées à l'intérieur, des compteurs peuvent y être déplacés partiellement. Et par conséquent, le premier alinéa de cet exemple, ne serait pas numéroté avec un, mais avec deux. Cependant, cela peut être facilement corrigé avec une réinitialisation (remise à zéro) rapide du compteur correspondant.

L'exemple est presque terminé, il manque encore la fin de l'environnement code :

```
\end{addmargin}
\end{document}
```

En réalité, il y avait aussi les autres paragraphes de la loi à commenter. Mais j'en ai décidé ainsi.

Cependant, attendez ! Et si, dans cet exemple, le paragraphe ne devait pas tenir sur la page ? Aurait-il ensuite passé la page suivante ? La réponse à cette question, traitée dans la prochaine section, est : il convient de rappeler qu'un argument optionnel vide n'est pas équivalent à un argument omis.

```
\syncwithnotecolumns[liste des noms de colonne de note]
```

Cette instruction effectue une concordance du texte principal avec tous les noms de colonnes de note séparés, dans une liste, par des virgules. Ici, le texte principal est coordonné avec la colonne dont la marque est, dans le document, la

plus en arrière. Cet effet secondaire permet de synchroniser les colonnes entre-elles.

Si l'argument n'est pas spécifié, est vide ou commence par `\relax`, il est synchronisé avec toutes les colonnes de notes déclarées.

## 18.6 Édition forcée des colonnes de notes

En plus de la sortie normale de la colonne de la note, comme décrit dans l'article 18.3, il est parfois nécessaire d'imprimer toutes les notes collectées qui ne sont pas encore émises. Ceci est particulièrement utile si les notes longues conduisent à de plus en plus de notes décalées vers le bas ou déplacées vers de nouvelles pages. Le bon moment pour une telle édition forcée est par exemple à la fin d'un chapitre ou à la fin du document.

```
\clearnotecolum[nom de la colonne de note]
```

Avec cette déclaration, toutes les notes d'une colonne déterminée, qui ne sont pas encore émises mais créées sur cette page ou une page précédente jusqu'à la fin de la page en cours, sont imprimées. L'édition des notes en attente nécessite la création obligatoire de pages blanches selon les besoins. Lors de l'impression des notes de cette colonne en attente, d'autres notes peuvent attendre, le cas échéant, aussi longtemps que nécessaire d'être émises avec les notes en attente de la colonne spécifiée.

Avec la mise en instance de notes peut se poser la question de notes émises à tort, insérées précédemment dans les marges de pages qui sont maintenant remplacées par des pages vierges. Ce point sera normalisé dans une prochaine amélioration de `LATEX`. Un tel changement entraîne généralement le message :

« `LATEX Warning : Label(s) may have changed. Rerun to get cross-references right.` » (`LATEX` attention : l'étiquette peut avoir changé. Recompiler pour obtenir des références croisées réelles).

Le nom de la colonne mémo dont les notes sont en attente d'impression est spécifié avec l'argument optionnel. Si aucun argument n'est spécifié, la colonne `marginpar` prédéfinie est utilisée.

Le lecteur attentif ne manquera pas de remarquer que la sortie forcée d'une colonne de note de la synchronisation n'est pas dissemblable. Dans ce cas, elle est située en haut de la page après la dernière édition et non pas directement en

dessous de cette dernière, ce qui permet de finaliser avec moins de recompilation de L<sup>A</sup>T<sub>E</sub>X.

```
\clearnotecolumns[liste de nom de notes de colonne]
```

Cette déclaration fonctionne d'une manière comparable à `\clearnotecolumn`. Mais ici, non seulement une colonne de note peut être spécifiée comme argument optionnel, mais une liste séparée par des virgules permet d'utiliser plusieurs noms de colonnes. Les notes en attente sont alors toutes issues de ces colonnes. Si l'argument n'est pas spécifié ou reste vide, les notes en attente de toutes les colonnes de notes sont diffusées.

```
autoclearnotecolumns = interrupteurs marche-arrêt
```

En règle générale, une note en attente peut toujours passer implicitement lorsque la commande `\clearpage` est exécutée à la suite de `\chapter`. C'est aussi possible à la fin d'un document avec `\end{document}`.

L'option `autoclearnotecolumns` permet de savoir, par contrôle automatique, si l'exécution de `\clearpage` ou de `\clearnotecolumns` est effectuée avec ou sans argument. Puisque l'on suppose que c'est généralement le cas, l'option est active par défaut, mais des valeurs adaptées pour les interrupteurs simples (voir la table 1.6 page 20), hors tension ou actifs au choix, peuvent être appliquées. Il convient de noter que dans le cas de désactivation de l'édition automatique, les notes en attente sont perdues, complètement ou partiellement, à la fin du document. Par conséquent, il faut utiliser, dans ce cas, `\clearnotecolumns` avant `\enddocument`.

Répondre maintenant à la question de l'exemple dans la section précédente : est-ce que le paragraphe présente une sortie complète ou se terminera à la page suivante ? ce qui serait, bien sûr, le cas par défaut. Et comme cela se passerait après la fin de l'environnement de `addmargin`, il pourrait encore se chevaucher avec le texte suivant. Par conséquent, il serait assez judicieux d'ajouter dans l'exemple un autre point de synchronisation après l'environnement `addmargin`. Voir le résultat de l'exemple dans la figure 21.1 page 579.

## 18.7 Extrait du manuel `sclayer-notecolumn`

L'idée de ce paq a été retenue après un contact e-mail avec Heiner RICHTER de FH STRALSUND, qui, pour un projet de livre, m'a demandé si une telle fonctionnalité pouvait être, pour moi, facilement ajoutée à un autre paq. Lui ne le pouvait pas, mais la conception et la mise en œuvre d'une version alpha de

test de `scrlayer-notecolumn` a été faite en trois heures environ. Alors peut-être que c'était vraiment une bonne idée de me poser la question.

Parfois, un document a besoin d'une colonne de notes supplémentaires à côté de la colonne de texte principal et peut-être à côté de la colonne marginale, et même d'avoir des sauts de page à l'intérieur de la colonne de notes. Ce paq fournit une solution basée sur l'interface de couche de `scrlayer`. Vous pouvez définir plusieurs colonnes de notes, chacune d'elles avec des sauts de page à l'intérieur. Il prend en charge le saut de page automatique à l'intérieur de notes, reconnaît les collisions qu'il peut traiter et résoudre par des notes en mouvement.

```
\DeclareNoteColumn[liste d'options]{note nom de colonne}
\DeclareNewNoteColumn[liste d'options]{note nom de colonne}
\ProvideNoteColumn[liste d'options]{note nom de colonne}
\RedeclareNoteColumn[liste d'options]{note}
```

`\DeclareNoteColumn` peut être utilisé pour déclarer une colonne de note. Une colonne de note couvre verticalement la zone de texte. `Horizontal` est, par défaut, la position ordinaire de la colonne de note. Pour produire des colonnes, utiliser `\reversemarginpar` et `\normalmarginpar`. Noter la différence avec la création d'une note de marge. La largeur de la colonne est par défaut `\marginparwidth`, mais vous pouvez utiliser un argument optionnel de la liste pour changer ce comportement.

La liste d'options est une liste de type `key=value` (clé=valeur) séparées par des virgules.

Actuellement, les clés suivantes sont supportées :

`font=font declaration`

- définit les attributs de la police initiale à utiliser pour la colonne de notes. Tout ce qui est autorisé et fixé par `\setkomafont` ou `\addtokomafont` peut être adopté comme déclaration de police. Noter que `\normalfont` et `\normalsize` sont employés avant et que vous n'en avez pas besoin pour votre initialisation.

`marginpar`

- met en place la position et la largeur à utiliser pour la colonne marginale de notes de `\marginpar`.

`normalmarginpar`

- met en place la position et la largeur à utiliser pour la colonne marginale normale de notes et ignore `\reversemarginpar` et `\normalmarginpar`.

`position=horizontal offset`

- crée le décalage horizontal de la colonne de note dans le bord gauche.

Ce peut être soit une longueur de L<sup>A</sup>T<sub>E</sub>X, une dimension ou une commande de T<sub>E</sub>X, une valeur de longueur, ou une expression de dimensions en utilisant « +, -, \*, / » et les parenthèses (plus d'informations sur les expressions dimensionnelles [Tea98], section 3.5). La valeur est calculée au moment de l'utilisation et non de la définition.

Ainsi, vous pouvez même utiliser des constructions plus complexes telle que :

```
\ifodd\value{page}
\oddsidemargin+1in
+\textwidth+\marginparsep
\else
\evensidemargin+1in
-\marginparsep
-\marginparwidth
\fi
```

`reversemarginpar`

- crée la position et la largeur pour utiliser la colonne inverse de note marginale et ignore `\reversemarginpar` et `\normalmarginpar`.

`width=largeur`

- met en place la dimension horizontale de la colonne de la note. Vous pouvez utiliser les mêmes valeurs pour la taille que pour le décalage horizontal de la position de l'option.

Vous pouvez utiliser des colonnes de notes d'autant que L<sup>A</sup>T<sub>E</sub>X fournit des boîtes de registres libres, parce que chaque colonne de la note en a besoin d'une. Utiliser le paq  $\varepsilon$ -T<sub>E</sub>X pour augmenter la quantité normale de boîtes de registres de 256 à 32768. Cela devrait être suffisant pour éviter l'erreur « pas de place pour `newbox` ». La mise en œuvre de colonnes de notes utilise des couches de premier plan du paq `sclayer`. Chaque `\DeclareNoteColumn` génère une couche et l'ajoute à la page de pseudo style `@everystyle@`. Théoriquement, mais bien que déconseillé, vous pouvez modifier la couche générée. Le chevauchement des colonnes de notes différentes est autorisé. Une colonne de note nommée `marginpar`, est prédéfinie.

Noter que changer la police initiale en utilisant

```
\setkomafont{marginnote} (nom colonne note){déclaration de police}
```



ou

`\addtokomafont{margin!marginnote}(nom colonne note){déclaration de police}` modifie la police de toutes les impressions de notes successives, mais pourrait rendre difficile la connaissance des notes affectées par le changement.

Voir section 3.6 plus d'informations sur `\setkomafont` et `\addtokomafont`.

La différence entre `\DeclareNoteColumn` et les autres commandes est que l'emploi de `\DeclareNoteColumn` définit la colonne de note indépendamment de l'existence d'une colonne de note portant le même nom.

`\DeclareNewNoteColumn` dénomme seulement les colonnes de notes dont les noms ne sont pas encore utilisés pour une autre colonne de note. Ce serait une erreur de l'utiliser avec un nom de colonne déjà attribué.

`\ProvideNoteColumn` est comme `\DeclareNewNoteColumn`, mais si le nom de colonne est déjà utilisé pour une colonne de note, rien ne se passe.

`\RedeclareNoteColumn` peut être utilisé pour déclarer une colonne de note avec un nom, qui est déjà utilisé. Son utilisation avec un nouveau nom serait une erreur.

<code>makenote[nom de colonne]{texte de la note}</code>
---

Cette commande est utilisée pour créer une nouvelle note qui sera écrite dans un fichier auxiliaire avec l'extension `.slnc` dans cette série et composée lors de la prochaine exécution, imprimée avec une position verticale utilisée pour une `\parbox[t]`, et déplacée horizontalement vers et dans la colonne. Les collisions sont reconnues dans cette colonne (mais pas dans d'autres colonnes). Dans un tel cas, la note peut être déplacée vers le bas et même vers une autre page. Les sauts de page automatiques dans les colonnes de note sont pris en charge et supportés. Ce fonctionnement diffère des classes de KOMA-Script `\marginpar` et `\marginline`, ou de `\marginnote` (paq `marginnote`).

Noter qu'après avoir changé le document, une double compilation de L<sup>A</sup>T<sub>E</sub>X est nécessaire pour déplacer la note à sa position finale.

Noter également que les commandes fragiles doivent être protégées pour éviter les messages d'erreur lors de l'écriture ou de la lecture du fichier auxiliaire.

Noter enfin que `scrlayer-notecolumn` n'utilise qu'un seul fichier auxiliaire pour l'ensemble de toutes les colonnes de note. Vous ne devriez pas éditer ou modifier ce fichier auxiliaire dont la syntaxe est fragile. Il ne sera pas lu à l'aide de `\input` mais en utilisant `\read`. Pour cette raison, les sauts de ligne, lignes vides et `\par` doivent être évités.

Actuellement on ne sait pas exactement où `makenote` peut être utilisé, mais surtout pas à l'intérieur de `makenote`. La valeur par défaut pour l'argument optionnel `makenote` est `marginpar`, c'est le nom prédéfini de la colonne de note.

```
\autoclearnotecolumns=simple interrupteur
\clearnotecolumn[nom de la colonne de note]
\clearnotecolumns[liste des noms de colonnes de note]
```

Les commandes peuvent être utilisées pour produire les notes intercalées dans une ou plusieurs colonnes de notes. Avec `\clearnotecolumn` seules les notes de la colonne avec le nom d'une colonne donnée seront éditées. Omettre l'argument optionnel, ce qui n'est pas la même chose qu'utiliser une valeur vide, entraîne l'utilisation de la colonne `marginnote`.

Avec `\clearnotecolumns`, les notes de toutes les listes de noms de colonnes séparées par une virgule seront éditées. Si vous omettez l'argument optionnel ou s'il est vide, ou s'il commence par `\relax`, les notes de toutes les colonnes de note seront éditées.

La macro `\clearpage` est utilisée pour l'édition des notes intégrées. Plusieurs compilations de  $\text{\LaTeX}$  peuvent être nécessaires afin de l'exécuter d'une façon correcte, pour l'ensemble du document. Vous obtiendrez le message :

«  $\text{\LaTeX}$  Warning : Label(s) may have changed » ( $\text{\LaTeX}$  attention : l'étiquette (s) peut avoir changé).

Recompiler pour obtenir des références croisées correctes aussi longtemps que le processus n'est pas réellement terminé.

Mais vous n'avez pas besoin, pour ce faire, d'utiliser `\clearnotecolum` ou `\clearnotecolumns` vous-même, parce que l'option `autoclearnotecolumns` est activée par défaut, ce qui signifie que `\clearnotecolumns` sera exécuté à chaque `\clearpage`. Vous pouvez changer cela en utilisant la valeur de commutation `off \autoclearnotecolumns` comme simple interrupteur. Noter que vous n'êtes pas autorisé à utiliser `\clearnotecolumn` ou `\clearnotecolumns` à l'intérieur d'une note. Vous ne pouvez utiliser ces commandes que dans le texte principal du document.

```
\syncwithnotecolumn[nom d'une colonne de note]
\sycwithnotecolumns[liste des noms de colonnes de note]
```

Ces commandes sont utilisées pour synchroniser la position verticale du texte principal avec une ou plusieurs colonnes de note. Utiliser l'argument optionnel de `\syncwithnotecolumn` donne le nom d'une colonne de note. S'il est omis,

`marginpar` est utilisé comme nom. Utiliser `\syncwithnotecolumn` correspond au même usage que `\syncwithnotecolumns` avec exactement le nom de colonne de la note.

L'argument optionnel de `\syncwithnotecolumns` consiste en une liste de noms séparés par des virgules. S'il est omis ou si vous utilisez un argument vide ou encore si l'argument commence par `\relax` la synchronisation se fera avec toutes les colonnes de notes déclarées.

La synchronisation avec une colonne de note signifie que la position courante du texte principal sera modifiée pour être la position verticale courante de la colonne de note après le nettoyage de cette colonne. Cela diffère dans plusieurs points du nettoyage d'une colonne de note

Après avoir nettoyé une colonne de note, la position du texte est changée pour constituer une nouvelle page, même si la colonne n'a pas rempli toute la page avant. La synchronisation va changer la position verticale juste après la dernière position verticale de la colonne. Le nettoyage d'une colonne nécessite moins de cycles de compilation de  $\text{\LaTeX}$  que la synchronisation d'une colonne.

L'un et l'autre, pour le nettoyage d'une synchronisation de colonne de note, utilisent le mécanisme d'étiquette de  $\text{\LaTeX}$  sans utiliser `\pageref`, `\label`, ou `\ref`, mais `\newlabel`. Donc, vous devriez jeter un œil aux messages de rediffusion de  $\text{\LaTeX}$  ayant comme sujet `label` non défini ou modifié.

L'utilisation de `\syncwithnotecolumn` ou de `\syncwithnotecolumns` n'est pas autorisée à l'intérieur d'une note, mais si vous le faites, vous devrez faire fonctionner  $\text{\LaTeX}$  indéfiniment sans aboutir à un état final. La synchronisation de plusieurs colonnes de notes dans le texte principal d'un document permet de coordonner les colonnes les unes avec les autres.



# Chapitre 19

## Informations complémentaires sur `typearea.sty`

Ce chapitre donne des informations supplémentaires sur le paq `typearea`. Certaines parties sont soumises à KOMA-Script livre [KM12]. Cela ne devrait pas être un problème pour l'utilisateur normal qui veut juste utiliser le paq et n'aura normalement pas besoin de ces informations.

D'autres informations sont destinées à l'utilisateur qui veut écrire son propre paq basé sur `typearea` ou confronté à des tâches inhabituelles.

Une partie de l'information examine les moyens `typearea` compatibles avec les classes `standard` ou des versions antérieures de KOMA-Script. Des pièces, fixées dans une police `sans serif`, existent seulement pour la compatibilité avec ces versions et ne doivent pas être utilisées.

### 19.1 Caractéristiques expérimentales

Dans cette section, les possibilités pragmatiques sont décrites.

Expérimentalement signifie dans ce contexte que la fonction ne peut pas être garantie, pour deux raisons :

- la fonction ou sa mise en œuvre finale n'est pas définitive,
- les possibilités dépendent en partie d'autres paqs internes et sont garanties aussi longtemps que ces autres paqs ne sont pas modifiés.

`usegeometry=on-off-valeur`

Normalement `typearea` se soucie peu de savoir s'il est utilisé en combinaison avec le paq `geometry` (voir [Ume10]). Cela signifie en particulier que `geometry` ne reçoit rien si vous changez, par exemple, le format de papier avec `typearea` - quelque chose que `geometry` ne propose pas.

Une fois que l'option `usegeometry` est réglée, `typearea` libère les options propres à la définition de `geometry`, et même de `\newgeometry` appelée lorsque de nouveaux paramètres sont activés dans le document (voir `\activateareas` dans la section suivante). Puisque `geometry` ne modifie pas l'orientation ou le format du papier via `\newgeometry`, cela est mis en œuvre par des instructions internes et longueurs de `geometry` et au besoin, fonctionne avec les versions de 5.3 à 5.6.

L'option ne signifie d'ailleurs pas que, lorsque vous êtes confronté à des tâches inhabituelles et que vous utilisez `geometry` pour un changement d'orientation ou de taille de papier papier avec `typearea`, le résultat soit utilisé directement de manière optimale. Pour des raisons de confort, `geometry` propose davantage d'options nécessaires pour le réglage du papier comme pour la détermination de la zone de texte, marges, entête, pied de page – appelées surdétermination – et en même temps de nouvelles informations manquantes de suppression déjà connues appelant `newGeometry` – on parle alors de maintien de la valeur – vous devez éventuellement déterminer complètement de nouvelles valeurs dans un appel à `newgeometry` séparé pour utiliser explicitement tous les paramètres souhaités. La prise en compte de `geometry` par `typearea` peut ouvrir des possibilités supplémentaires.

`areasetadvanced=on-off-valeur`  
`\areaset [BCOR] {largeur}{hauteur}`

Normalement on considère les options `areaset` pour déterminer la hauteur de l'entête et du pied de page ou pour déterminer si la partie limite d'élément de la zone de type, devrait être réglée non pas de la même manière que `typearea`. avec l'option `areasetadvanced` mais que `areaset` se comporte à cet égard plus comme `typearea`. Néanmoins les paramètres diffèrent, qui conduisent à de grands champs continus de texte entre les deux commandes parce que `typearea` est toujours arrondi à la ligne entière en appliquant si besoin le bord inférieur jusqu'à une ligne sélectionnée plus petite tandis que les bordures `areaset`, haut et bas, sont toujours établies dans une proportion 1 : 2. Les zones de texte des différentes commandes peuvent par conséquent être facilement déplacées verticalement avec la même taille.

## 19.2 Commandes pour experts

Dans cette section, les instructions décrites n'ont que peu ou pas d'intérêt pour l'utilisateur moyen, mais offrent des fonctionnalités supplémentaires pour les experts, d'une manière condensée.

`\activateareas`

Cette déclaration est utilisée par `typearea` pour transférer les paramètres de mise en page et de marges correspondant dans les longueurs internes de L<sup>A</sup>T<sub>E</sub>X chaque fois que la zone de type est recalculée à l'intérieur du document, c'est-à-dire, après `\begin{document}`. Si l'option `\pagesize` est utilisée à nouveau, elle sera réexécutée avec la même valeur. La taille de page peut varier, par exemple dans un document au format PDF. L'expert peut utiliser ces instructions s'il modifie `\textwidth` et `\textheight` manuellement dans le document. Il est responsable des sauts de page éventuellement nécessaires avant ou après l'utilisation de `\activateareas` dont toutes les modifications sont locales.

```
\storeareas{commande}
\BeforeRestoreareas{code}
\BeforeRestoreareas{code}
\AfterRestoreareas{code}
\AfterRestoreareas*{code}
```

Les macros `\storeareas` permettent de définir tous les paramètres actuels qui peuvent être utilisés pour restaurer la zone de frappe et les marges. Ainsi, il est possible d'enregistrer les paramètres actuels, de les modifier, de les utiliser modifiés, puis de restaurer les paramètres d'origine.

Vous voulez, dans un document au format portrait, adapter une page en mode paysage. Pas de problèmes en utilisant `\storeareas` :

```
\documentclass[\texttt{pagesize}]{scrartcl}
\begin{document}
\noindent\rule{\textwidth}{\textheight}
\storeareas\mesvaleursenregistrées
\KOMAOPTIONS{paper=landscape,\cott{DIV=current}}
\noindent\rule{\textwidth}{\textheight}
\clearpage
\mesvaleursenregistrées
\noindent\rule{\textwidth}{\textheight}
\end{document}
```

Il est indispensable d'appeler `\mesvaleursenregistrées` avant de déclarer `\clearpage`, la restauration s'effectue uniquement sur la page suivante.

Les documents recto-verso devraient apporter des modifications dans la taille du papier. Au lieu de cela, même `\cleardoubleoddpage` ou, si aucune classe KOMA-Script n'est employée, `\cleardoublepage` peut être utilisé.

Dans l'exemple, `\noindent` est utilisé pour prévenir le retrait de paragraphe normal avant les boîtes noires. Mais il reflète pas une image correcte de la mise en page.

Lorsque vous utilisez `\storeareas` il est à noter que les deux zones de stockage ainsi que les instructions pour les définir ne devraient pas être appelées dans un groupe. La définition des instructions est faite en interne avec `\newcommand`. À la réutilisation d'une instruction déjà définie un message d'erreur est émis.

<pre>\AfterCalculatingTypearea{instructions} \AfterCalculatingTypearea*{instructions} \AfterSettingArea{instructions} \AfterSettingArea*{instructions}</pre>
--

Ces instructions sont utiles pour l'administration de deux crochets. Les deux premières, `\AfterCalculatingTypearea` et sa forme étoilée permettent aux experts de recalculer une nouvelle division dans l'espace de l'impression et des marges, à chaque fois, après toute version implicite ou explicite de `typearea`. Les deux suivantes, `AfterSettingArea` et sa forme étoilée, sont utilisées pour exécuter `areaset`. Les formes normales de travail affectent l'ensemble, tandis que les modifications apportées par les formes étoilées sont seulement locales.

Les instructions sont exécutées immédiatement avant `\activateareas`.

## 19.3 Fichiers `typearea.cfg` de paramètres locaux

Pour plus d'informations, consulter le livre KOMA-Script [KM12].

## 19.4 Options plus ou moins obsolètes

Pour plus d'informations, consulter le livre KOMA-Script [KM12]



# Chapitre 20

## Informations supplémentaires sur les classes

Ce chapitre fournit des précisions sur les classes de KOMA-Script `scrbook`, `scrreprt` et `scrartcl`, mais aussi des instructions concernant des fonctions pour le paq `scrxextend`. Certaines parties de ce chapitre ont trait à KOMA-Script book [KM12]. Cela n'est pas un problème pour un utilisateur qui applique juste les classes et n'a pas besoin de ces informations. Certaines sont à destination de l'utilisateur qui résout des tâches inhabituelles ou veut écrire ses propres classes en utilisant KOMA-Script. Une autre partie de l'information concerne l'harmonie de ces classes avec les classes L<sup>A</sup>T<sub>E</sub>X standard ou des versions existantes antérieures de KOMA-Script. Des pièces existent seulement pour la compatibilité avec les versions antérieures de KOMA-Script. Elles sont écrites dans une police sans serif et ne doivent plus être utilisées.

Vous trouverez à ce sujet plus d'informations dans le livre KOMA-Script [KM12].

### 20.1 Compléments d'information à propos du mode d'emploi

Vous trouverez à ce sujet des d'informations dans le livre KOMA-Script [KM12].

## 20.2 Interaction avec d'autres paqs

Vous trouverez à ce sujet des informations dans le livre KOMA-Script [KM12].

## 20.3 Instructions pour experts

Dans cette section, les instructions décrites ont, pour l'utilisateur moyen, peu ou pas d'intérêt, mais ces commandes offrent des possibilités supplémentaires pour les experts et puisque l'information leur est principalement destinée, elle est condensée.

```
\KOMAClassName \ClassName
```

`\KOMAClassName` stocke le nom de la classe KOMA-Script actuelle utilisée. Quelqu'un qui veut savoir quel KOMA-Script ou quelle classe est utilisé peut facilement le tester avec cette commande. En revanche, `\ClassName` indique la classe `standard` qui est remplacée par la classe KOMA-Script. L'existence de KOMA-Script n'indique pas forcément l'utilisation d'une classe KOMA-Script. Non seulement KOMA-Script mais aussi tous ses paqs définissent KOMA-Script, et d'autres paqs peuvent juger utile de définir le pictogramme KOMA-Script avec ce nom.

```
\addtoentrydefault{niveau}{numéro}{titre}
```

Pour créer une table des matières, la classe KOMA-Script v3.08 n'utilise pas `\addcontentsline` directement, mais appelle `\addtoentrydefault` avec des arguments similaires. La commande peut être utilisée à la fois pour les entrées numérotée ou non.

Ainsi chaque niveau correspond à celui de l'organisation du texte, c'est-à-dire `partie`, `chapitre`, `section`, `subsection`, `subsubsection`, `paragraphe` ou `alinéa`. Le numéro des sections déjà formatées est donné par le deuxième argument qui peut être vide. Le texte de l'entrée dépend de l'argument rubrique. Il est recommandé de protéger les commandes fragiles dans cet argument avec `\protect`. Une fonction spéciale s'applique au numéro (*number*) de l'argument : un argument vide signale qu'une entrée non numérotée doit être générée. Par défaut, le pré-réglage sera réalisé avec :

```
\addcontentsline{toc}{niveau}{\%  
\protect\numberline titre\%  
}
```

Toutefois, si l'argument n'est pas vide une entrée numérotée sera générée dont le numéro sera celui de l'article préformaté.

Par défaut, utiliser KOMA-Script :

```
\addcontentsline{toc}{niveau}{\%
\protect \numberline{numéro}titre\%
}
```

pour la génération de cette entrée. Les auteurs de paqs et de classes `Wrapper` peuvent redéfinir cette déclaration pour manipuler les entrées. Ainsi, on pourrait suggérer par exemple :

```
\renewcommand{\addtocentrydefault}[3]{\%
\ifstr{#3}{}{\%
}{\%
\ifstr{#2}{}{\%
\addcontentsline{toc}{#1}{#3}\%{\protect\numberline#3}\%
}{\%
\addcontentsline{toc}{#1}{\protect\numberline{#2}#3}\%
}{\%
}{\%
}\%
}
```

indispensable pour s'assurer que les articles n'ont pas d'entrée avec un titre vide. Dans la pratique, un tel changement n'est pas nécessaire car la suppression des entrées vides est déjà traitée d'autres manières dans les classes de KOMA-Script. Voir la description des structures de commandes dans la [section 3.16](#).

```
\addparttocentry{numéro}{titre}
\addchaptertocentry{numéro}{titre}
\addsectiontocentry{numéro}{titre}
\addsubsectiontocentry{numéro}{titre}
\addsubsubsectiontocentry{numéro}{titre}
\addparagraphtocentry{numéro}{titre}
\addsubparagraphtocentry{numéro}{titre}
```

La déclaration `\addtocentrydefault` est appelée directement si aucune commande individuelle n'a été définie pour le niveau spécifié ou si cette commande est `\relax`. Les instructions données sont fixées pour transmettre leur niveau et les arguments directement à `\addtocentrydefault`.

```
\raggedchapterentry
```

Il est possible, avec `\raggedchapterentry`, de définir le contenu du résumé

d'une table de chapitre avec un réglage préjustifié. Pour cela, il suffit simplement de définir `\raggedright`. Voir aussi les restrictions dans la section 21.2. Par défaut, la définition est vide. D'autres paramètres `\raggedchapterentry` ne sont pas recommandés et peuvent provoquer des résultats inattendus.

```
\@fontsizefilebase
\changefontsizes{taille de police}
```

Le préfixe `scrsiz` spécifié pour les tailles de police dans la section 21.1 est simplement le réglage par défaut de la macro interne `\@fontsizefilebase` utilisée lorsque la macro n'a pas été définie lors du chargement d'une classe KOMA-Script ou du paq `scrextend`. Les auteurs de classes `Wrapper` peuvent définir cette macro avec un autre préfixe pour utiliser différemment les fichiers de tailles de police. Ils peuvent aussi modifier ou désactiver la solution de repli en redéfinissant `\changefontsizes`. Cette macro supporte un seul argument : la taille de police désirée.

```
\newkomafont[avertissement]{article}{par default}
\aliaskomafont{nom d'alias}{article}
```

Les experts peuvent utiliser `\newkomafont` afin de définir, par défaut, une valeur pour le style de police d'un élément. Par la suite, ce « par défaut » peut être modifié à l'aide des commandes `\setkomafont` et `\addtokomafont` (voir section 3.6). Bien sûr, ce n'est pas assez pour user du style de police défini. L'expert lui-même doit préparer son code pour utiliser la commande `\usekomafont` et l'appliquer dans la définition des codes de cet élément.

L'argument optionnel définit un message d'avertissement affiché chaque fois que le style de police par défaut de cet élément sera changé. Dans ce cas, l'expéditeur de l'avertissement sera la classe de KOMA-Script utilisée ou le paq `scrextend`. La commande `\aliaskomafont` définit un `alias` pour un élément existant déjà défini.

KOMA-Script informera l'utilisateur automatiquement sur le véritable nom d'élément. Un nom `d'alias` peut être utilisé, par exemple, si un développeur trouve un meilleur nom pour un élément qui a été défini précédemment avec un autre nom et si cet ancien nom doit rester utilisable en raison de la compatibilité. Aussi un `alias` peut augmenter une facilité d'utilisation. KOMA-Script lui-même fait une riche utilisation de cette opportunité.

```
\addtokomafontrelaxlist{macro}
\addtokomafontgobblelist{macro}
```

Comme il est expliqué dans la première partie de ce manuel, les paramètres

des éléments de police ne contiennent que des commandes de sélection de taille, famille, codage, épaisseur de ligne, forme et couleur. Un changement de couleur, dans L<sup>A</sup>T<sub>E</sub>X, n'est pas transparent et peut donc provoquer des effets indésirables si `\usekomafont` est utilisé à un emplacement défavorable.

Actuellement, les utilisateurs ont tendance à inclure, dans les paramètres de police, des éléments disparates parfois très critiques, tels que `\MakeUppercase` en fin de réglage.

Beaucoup de ces paramètres, théoriquement interdits, travaillent en interne, sans difficulté ni anomalie véritable, même si la dernière commande dans le réglage d'un argument de police doit, par exemple, utiliser `\textbf` au lieu de `\bfseries`. Naturellement, il n'y a aucune garantie.

Dans certains cas, il est nécessaire de limiter les paramètres de police au sein de KOMA-Script et on utilisera `\usefontofkomafont` plutôt que `\usekomafont` (voir la section 3.6).

Les commandes `\usefontofkomafont` et apparentées ont leurs limites. Par conséquent, les paramètres de police présumés d'un élément ne doivent pas s'attendre à un argument entièrement expansif, ce qui est le cas, par exemple, dans `\MakeUppercase`. KOMA-Script contient une liste interne de macros avec `\usefontofkomafont` qui devrait être à `\relax` et qui comprend par défaut `\normalcolor`, `\MakeUppercase` et `\MakeLowercase`. Individuellement, d'autres instructions peuvent être ajoutées à `\adatokomafontrelaxlist`.

Il convient de noter que la macro spécifiée est vraiment définie pour `\relax`. Les arguments dans le cadre de la police sont donc réalisés localement et des déclarations telle que `\@setlength` ne sont jamais ajoutées à la liste. Vous êtes responsable des erreurs causées par l'utilisation de `\adatokomafontrelaxlist`. En outre, cette possibilité ne doit être interprétée comme une légitimation des paramètres de police et permettre d'ajouter toutes sortes d'instructions. Mais si une macro au sein de `\usefontofkomafont` ou apparenté comprend un seul argument oublié, il est alors préférable d'utiliser `\adatokomafontgobblelist`. Un exemple est la déclaration `\color` qui doit être ignorée, y compris le nom de la couleur et donc figure déjà dans la partie par défaut de cette liste.

`\IfExistskomafont{élément}{code}{sinon}`

Depuis les écritures, quelques éléments peuvent être modifiés uniquement à partir de certaines versions de KOMA-Script, il est parfois utile de pré-tester un élément pour voir si cette possibilité existe. L'instruction exécute alors le code si, et seulement si, l'article était, via `\newkomafont` ou `\aliaskomafont`, défini et donc changé avec `\setkomafont` ou `\adatokomafont` et si l'instruction

`\use ... komafont` peut être interrogée. Dans le cas contraire, le code `sinon` est exécuté.

```
\setparsizes{tiret}{distance}{ligne de fin d'espace vide}
```

Cette déclaration permet, à la fois, le retrait, l'espacement et l'ajustement de l'espace libre à la fin de la dernière ligne de chaque paragraphe de `scrbook`, `scrcrpt` ou `scrartcl`. Cette commande doit être utilisée chaque fois que des changements devraient aussi être reconnus par l'option `parskip=relative`. KOMA-Script lui-même l'utilise, par exemple, avec :

```
\setparsizes{0pt}{0pt}{0pt plus 1fil}
```

pour désactiver le retrait de paragraphe, la distance entre paragraphes ainsi que pour laisser un espace blanc à la fin de la dernière ligne de paragraphe. Une telle mesure est utile si un paragraphe ne se compose que d'une boîte, qui est réglée sans écart vers le haut ou vers le bas et occupe toute la largeur de la colonne. Si la boîte ne doit pas s'étendre sur toute la largeur et doit être imprimée avec les paramètres actuels de retrait et d'espace entre les paragraphes, l'utilisation de :

```
\setlength{\parfillskip}{0pt plus 1fil}
```

serait préférable et recommandée.

Un nouveau calcul ou la réactivation des paramètres pour l'empage et les marges (voir le chapitre 2) conduit, depuis KOMA-Script, à un réajustement à travers `\setparsizes` pour définir les valeurs si celles-ci n'ont pas changé dans l'intervalle. Cela devrait être une raison de plus pour ne pas changer les réglages passés de KOMA-Script. Les paramètres de recalcul sont compatibles avec les versions antérieures (voir section 3.2, versions en option) désactivées. Plus d'informations dans [Koh14a].

```
\DeclareSectionCommand[paramètres]{nom}
\DeclareNewSectionCommand[paramètres]{nom}
\RedeclareSectionCommand[paramètres]{nom}
\ProvideSectionCommand[paramètres]{nom}
```

Ces instructions décrivent une nouvelle commande `\name` qui peut être définie ou une commande `\name` existante qui peut être modifiée. Les réglages sont effectués en utilisant l'argument optionnel. Les paramètres sont séparés par une virgule, sous la forme d'une liste `clé=valeur`. En plus des caractéristiques indépendantes du style de titre qui se trouvent dans la table 21.1 page 574 existent des propriétés qui dépendent de chaque style particulier. Actuellement, les modèles suivants sont disponibles.

`\DeclareNewSectionCommand` est utilisé pour définir une nouvelle commande de la division. Si le nom est déjà utilisé par `TEX`, une erreur est émise sans aucune redéfinition à la place.

`\ProvideSectionCommand` se comporte de manière similaire, mais ne génère aucune message d'erreur..

`\RedeclareSectionCommand` cependant, ne peut être utilisé pour modifier une commande cadre existante avec les propriétés spécifiées, mais ne vérifie pas si la commande `\nom` existait auparavant. Ce nom doit être unique pour `TEX`.

Avec `\DeclareSectionCommand`, `TEX` ne vérifie pas si le `nom` est déjà utilisé et la commande `\nom` est strictement définie avec les propriétés spécifiées.

- `chapter` : style d'un titre de chapitre utilisé par défaut pour `\chapter` et indirectement pour `\addchap`. Des rubriques peuvent être définies dans ce style, qui soutient automatiquement une variante `\add...` Pour former les rubriques existantes ou nouvelles, voir la table 20.3 page 558. `\addchap` ainsi que les formes étoilées sont automatiquement refaçonnées avec `\chapter` et ne peuvent être modifiées indépendamment. Noter que ces styles de `scrartcl` ne sont pas fournis.
- `part` : style de la sous-rubrique utilisée pour `\part` et indirectement pour `\addpart`. De nouvelles rubriques dans ce style qui soutient une variante `\add...` peuvent être définies. Pour configurer les rubriques existantes ou nouvelles, voir la table 20.2 page 557. La commande `\addpart` ainsi que les formes étoilées sont automatiquement reconfigurés avec `\part` et ne peuvent pas être modifiées indépendamment.
- `section` : style du titre actuel de `section` utilisé, `\section`, `\subsection`, `\subsubsection`, `\paragraph` et `\subparagraph`. De nouvelle rubriques peuvent être définies dans ce style. Pour la configuration des rubriques existantes ou nouvelles et les propriétés des entêtes voir la table 21.3 page 598 Après une redéfinition, les touches de style `font`, `level`, `indent`, `tocindent`, `afterskip`, `beforeskip` et `tocnumwidth` sont obligatoires. Ceci vaut également si une commande qui n'était pas une commande de division est redéfinie comme telle avec `\RedeclareSectionCommand`. La commande `\addsec` est reconfigurée automatiquement avec `\section`, comme les formes étoilées et ne peut pas être modifiée à part. Lorsqu'une commande de sectionnement est définie dans ce style, un élément éponyme dont les paramètres de police peuvent être modifiés avec `\setkomafont` et `\addtokomafont` est créé simultanément (voir la `section` 3.6), s'il n'existe pas encore.

Chaque division comprend également une contre-déclaration du même nom, créée avec `\newcounter` intégrant, si nécessaire, quatre instructions. Il en va de même pour la sortie du compteur : `\thenom`, son format, `\nomformat` pour créer un titre de colonne, `\nommark`, le formatage du compteur : `\nommarkformat`, le nom de l'objet et le niveau de contour `\nomnumdepth`. L'instruction `est`, le cas échéant, prédéfinie pour créer un titre de colonne `\nommark` de sorte qu'aucun titre de colonne n'est généré. La sortie du compteur, `\thenom`, est prédéfinie pour utiliser les chiffres arabes. Lorsque la clé définie à l'intérieur du compteur dépend d'un autre compteur, le résultat produit s'affiche en premier, séparé par un point.

Si pour une raison quelconque, `\paragraph` ne doit plus être redéfini avec des références, mais comme une rubrique telle que `\subsubsection` qui sera insérée dans l'entête à une distance de 10pt et sous la rubrique, mais sans aucune distance supplémentaire, il est possible d'utiliser :

```
\RedeclareSectionCommand[\%
beforeskip=-10pt,\%
afterskip=1sp\%
]{paragraph}
```

La distance verticale négative au-dessus du titre est produite avec `beforeskip` arrêtant la collecte de la première `section` après l'intitulé, et en toute logique, aucun espace vertical n'est souhaité après le titre, ce qui est signalé ici par la valeur `1sp`, `0pt` n'est pas considéré par  $\text{\LaTeX}$ , à ce stade, comme une valeur positive et génère un titre sous forme de référence. La valeur positive la plus petite est `1sp`. En général, il est préférable, pour la compensation verticale (voir `\flushbottom`, [section 3.4](#)), que les distances disposent d'une marge de manœuvre extensible :

```
\RedeclareSectionCommand[\%
beforeskip=-10pt plus -2pt minus -1pt,\%
afterskip=1sp plus -1sp minus 1sp\%
]{paragraph}
```

Dans cette application, la séparation des mots, utilisée comme repère des changements de distance verticale, devient négative avec `beforeskip` donné dans l'exemple, mais dans le même temps, il est possible de l'ignorer quand la distance nécessaire se rétrécit effectivement jusqu'à 0.

Dans cet exemple, seules les clés `beforeskip` et `afterskip` sont utilisées, et ce depuis que, dans `KOMA-Script v3.15`, `\paragraph` est déjà défini en interne avec `\DeclareSectionCommand`, donc que les autres paramètres peuvent être



adoptés sans changement. La définition originale de `\paragraph` correspond à celle de `scrartcl` :

```
\DeclareSectionCommand[\%
level=4,
indent=0pt,
beforeskip=3.25ex plus 1ex minus .2ex,
afterskip=-1em,
font={},
tocindent=7em,
tocnumwidth=4.1em,
counterwithin=subsubsection
]{paragraph}
```

Dans `scrreprt` et `scrbook` sont parfois utilisées des valeurs dissidentes. Dans `\chapter` des paramètres dépendent de l'option rubriques (voir section 3.16).

Ces paramètres spécifiques sont indiqués dans la table 20.3 page 558. La table 20.5 page 560 fournit un aperçu de tous les préréglages. Il convient de noter que l'enregistrement de `1ex` et `\baselineskip` dans la taille prédéfinie de l'entête dépend de l'entrée dans la table des matières.

TABLE 20.1 – Styles de titre et valeurs des propriétés déclarées dans le dispositif de commandes

Clé	Valeur	Signification
<code>counterwithin</code>	nom du compteur	l'appartenance à l'indicateur de niveau dépend de la valeur fixée par le compteur. Si elle augmente via <code>\stepcounter</code> ou <code>\refstepcounter</code> , alors le niveau hiérarchique pour le compteur associé est remis à 0. En outre <code>\thecountername</code> suivi d'un point à la sortie du niveau de détail associé au compteur est préfixé.
<code>counterwithout</code>	nom du compteur	annule un réglage précédent reconnu au sein du compteur et donc seulement utilisé pour une modification des commandes existantes

Clé	Valeur	Signification (suite)
<code>expandtopt</code>	interrupteur	si cette option est activée, les valeurs de réglage des longueurs indiquées ci-dessous sont développées, évaluées et stockées, converties en pt. Sinon, toutes les valeurs indiquées ci-dessous pour les longueurs le sont uniquement à des fins de test, évaluées mais non enregistrées. Elle comprend les valeurs de simples interrupteurs de la table 1.6 page 20,
<code>level</code>	entier	valeur numérique hiérarchique (voir compteur <code>secnumdepth</code> , section 3.16), la valeur doit être unique.
<code>style</code>	nom	indique le style de rubriques
<code>tocstyle</code>	nom	Indique le style d'entrées de répertoire. Tous les styles de la liste des répertoires définis précédemment peuvent être utilisés (voir la section 14.3). Un nom vide empêche la définition de la commande pour les entrées de répertoire.
<code>tocOption</code>	valeur	D'autres options par l'intermédiaire de <code>tocstyle</code> sélectionnent le style d'entrée de répertoire. Voir la section 14.3. Pour les styles prédéfinis d'entrée, on trouve les attributs utilisables en option dans la table 14.1 page 442.

\subsubsection	
Réglage	Valeur prédéfinie
<code>afterskip</code>	1.5ex plus .2ex
<code>beforeskip</code>	-3.25ex plus -1ex minus -.2ex
<code>font</code>	voir l'élément <code>subsubsection</code> , table 3.15 page 208
<code>indent</code>	0pt
<code>level</code>	3
<code>tocindent</code>	7.0em
<code>tocnumwidth</code>	4.1em

TABLE 20.2 – Clés supplémentaires et valeurs des propriétés dans la déclaration de commandes du style `section`

Clé	Valeur	Signification
<code>afterskip</code>	longueur	une valeur négative signifie que le texte d'entête est utilisé comme référence. Le montant indique, dans ce cas, la distance horizontale après le titre. Une valeur positive conduit à une distance verticale après l'entête.
<code>beforeskip</code>	longueur	le montant indique la distance horizontale avant l'entête. Une valeur négative peut, néanmoins, insérer un espace positif. Les valeurs négatives signifient que le paragraphe devrait être placé après l'entête sans aménagement horizontal.
<code>font</code>	réglage de la police	paramètres de police utilisés dans l'intitulé de l'édition, en plus de l'élément disposition. Tous les paramètres permis à <code>\setkomafont</code> et <code>\adDTokomafont</code> pour l'élément commandes sectionnelles sont autorisés.
<code>indent</code>	longueur	Indentation de la marge de gauche avant le nombre et le texte de la rubrique.

`\paragraph`

Réglage	Valeur prédéfinie
<code>afterskip</code>	-1em
<code>beforeskip</code>	3.25ex plus 1ex moins .2ex
<code>font</code>	voir l'élément <code>paragraph</code> , table 3.15 page 208
<code>indent</code>	0pt
<code>level</code>	4
<code>tocindent</code>	10em
<code>tocnumwidth</code>	5em

TABLE 20.3 – Clés supplémentaires et valeurs des propriétés dans la répartition de la configuration des commandes du style chapter

Clé	Valeur	Signification
<code>afterskip</code>	longueur	le montant indique la distance verticale après l'entête.
<code>beforeskip</code>	longueur	le montant indique la distance horizontale avant l'entête. Une valeur négative peut, néanmoins, insérer un espace positif. Les valeurs négatives signifient que le paragraphe devrait être placé après l'entête sans aménagement horizontal.
<code>font</code>	réglage de la police	paramètres de police utilisés dans l'entête, en plus de l'élément <code>disposition</code> . Tous les paramètres permis à <code>\setkomafont</code> et <code>\addtokomafont</code> pour l'élément commandes sectionnelles sont autorisés.
<code>innerskip</code>	longueur	distance verticale entre le préfixe et le texte de l'entête, si un préfixe est utilisé.
<code>pagestyle</code>	style de page	nom du style de page à utiliser pour la page avec l'entête, sans vérification de validité. Des informations erronées provoqueront par conséquent, des messages d'erreur lors de l'utilisation de la commande de sectionnement.
<code>prefixfont</code>	réglage de la police	paramètres de police, en plus de l'élément de planification et de l'élément des commandes sectionnelles pour être utilisés lors de la délivrance d'une ligne de préfixe dans l'entête. Tous les paramètres permis à <code>\setkomafont</code> et <code>\addtokomafont</code> pour l'élément commandes sectionnelles sont autorisés.

`\subparagraph`

Réglage	Valeur prédéfinie
<code>afterskip</code>	-1em
<code>beforeskip</code>	3.25ex plus 1ex moins .2ex
<code>font</code>	voir l'élément <code>subparagraph</code> , table 3.15 page 208
<code>indent</code>	<code>\scr@parindent</code>
<code>level</code>	5
<code>tocindent</code>	12em
<code>tocnumwidth</code>	6em

TABLE 20.4 – Clés supplémentaires et valeurs des propriétés dans la répartition de la configuration des commandes du style `part`

Clé	Valeur	Signification
<code>afterskip</code>	longueur	le montant indique la distance verticale après l'entête.
<code>beforeskip</code>	longueur	le montant indique la distance horizontale avant l'entête. Une valeur négative peut, néanmoins, insérer un espace positif. Les valeurs négatives signifient que le paragraphe devrait être placé après l'entête sans aménagement horizontal.
<code>font</code>	réglage de la police	paramètres de police utilisés dans l'entête, en plus de l'élément <code>disposition</code> . Tous les paramètres permis à <code>\setkomafont</code> et <code>\addtokomafont</code> pour l'élément commandes sectionnelles sont autorisés.
<code>innerskip</code>	longueur	distance verticale entre le préfixe et le texte de l'entête, avec <code>scrbook</code> et <code>scrreprt</code> .
<code>vpagestyle</code>	style de page	nom du style de page à utiliser pour la page avec l'entête, sans vérification de validité. Des informations erronées provoqueront par conséquent des messages d'erreur lors de l'utilisation de la commande de sectionnement. Cette possibilité n'existe qu'avec <code>scrbook</code> et <code>scrreprt</code> .
<code>prefixfont</code>	réglage de la police	paramètres de police, en plus de l'élément de planification et de l'élément des commandes sectionnelles pour être utilisés lors de la délivrance d'une ligne de préfixe dans l'entête. Tous les paramètres permis à <code>\setkomafont</code> et <code>\addtokomafont</code> pour l'élément commandes sectionnelles sont autorisés.

TABLE 20.5 – Préférences pour les rubriques de `scrbook` et `scrreprt`  
`\part`

Réglage	Valeur prédéfinie
<code>afterskip</code>	0pt plus 1fil
<code>beforeskip</code>	0pt plus 1fil + <code>\baselineskip</code>
<code>vfont</code>	voir l'élément <code>part</code> , table 3.15 page 208
<code>innerskip</code>	20pt
<code>level</code>	-1
<code>prefixfont</code>	voir l'élément <code>partnumber</code> , table 3.15 page 208.
<code>tocindent</code>	0pt
<code>tocnumwidth</code>	2em

`\chapter`

Réglage	Valeur prédéfinie
<code>afterskip</code>	voir la table 21.5 page 599
<code>beforeskip</code>	voir la table 21.5 page 599
<code>font</code>	voir l'élément <code>chapter</code> , table 3.15 page 208
<code>innerskip</code>	0pt
<code>level</code>	0
<code>prefixfont</code>	voir l'élément <code>chapterprefix</code> , table 3.15 page 208
<code>tocindent</code>	0pt
<code>tocnumwidth</code>	1.5em

Avec `headings=normal`

Réglage	Valeur prédéfinie
<code>afterskip</code>	<code>1.5\baselineskip</code> plus <code>.1\baselineskip</code> minus <code>.167\baselineskip</code>
<code>beforeskip</code>	<code>3\baselineskip+\parskip</code>
<code>font</code>	<code>\LARGE</code>

<code>\section</code>	
Réglage	Valeur prédéfinie
<code>afterskip</code>	2.3ex plus .2ex
<code>beforeskip</code>	-3.5ex plus -1ex minus -.2ex
<code>font</code>	voir l'élément <code>section</code> , table 3.15 page 208
<code>indent</code>	0pt
<code>level</code>	1
<code>tocindent</code>	1.5em
<code>tocnumwidth</code>	2.3em

<code>\subsection</code>	
Réglage	Valeur prédéfinie
<code>afterskip</code>	1.5ex plus .2ex
<code>beforeskip</code>	-3.25ex plus -1ex minus -.2ex
<code>font</code>	voir l'élément <code>subsection</code> , table 3.15 page 208
<code>indent</code>	0pt
<code>level</code>	2
<code>tocindent</code>	3.8em
<code>tocnumwidth</code>	3.2em

TABLE 20.6 – Préférences pour les entêtes de chapitre de `scrbook` et `scrreprt` avec l'option `headings`Avec `headings=big`

Réglage	Valeur prédéfinie
<code>afterskip</code>	$1.725\backslash\text{baselineskip}$ plus $.115\backslash\text{baselineskip}$ minus $.192\backslash\text{baselineskip}$
<code>beforeskip</code>	$3.3\backslash\text{baselineskip}+\backslash\text{parskip}$
<code>font</code>	<code>\huge</code>

```

\DeclareSectionCommand[paramètres]{liste de noms}
\DeclareNewSectionCommand[paramètres]{liste de noms}
\RedeclareSectionCommand[paramètres]{liste de noms}
\ProvideSectionCommand[paramètres]{liste de noms}

```

Ces instructions décrivent un certain nombre de commandes qui peuvent être définies ou modifiées. Dans cette liste, les noms des commandes de sectionne-

Avec <code>headings=small</code>	
Réglage	Valeur prédéfinie
<code>afterskip</code>	<code>1.35\baselineskip</code> plus <code>.09\baselineskip</code> minus <code>.15\baselineskip</code>
<code>beforeskip</code>	<code>2.8\baselineskip+\parskip</code>
<code>font</code>	<code>\Large</code>

ment sont séparées par des virgules. Ces instructions diffèrent sur deux points de l'énoncé expliqué précédemment pour définir ou modifier une commande de sectionnement unique.

Quand une déclaration `\DeclareNewSectionCommands` existait auparavant ou n'existait pas encore, la définition est réalisée avec `\RedeclareSectionCommands` mais une erreur appropriée est signalée.

Deuxièmement, il y a un autre paramètre, `increaselevel=entier`. Ainsi les modifications des valeurs de `level` et `toclevel` (voir la table 21.1 page 574), ne sont que des paramètres de la première commande de sectionnement à partir de la liste des noms. Pour toutes les autres commandes, les valeurs de `level` et `toclevel` augmentées de la valeur de `increaselevel` qui est de 1 par défaut, seront utilisées.

```
\newcommand*{\chapterheadstartvskip}{\vspace}{\@tempskipa}
\newcommand*{\chapterheadmidvskip}{\par\nobreak\vskip\@tempskipa}
\newcommand*{\chapterheadendvskip}{\vskip\@tempskipa}
```

Ceux-ci sont réactivées à l'aide des options `headings=big`, `headings=normal` ou `headings=small` à chaque fois. L'instruction `\chapter` définit la longueur interne `\@tempskipa` avant d'appeler `\chapterheadstartvskip` à la valeur qui résulte de l'application de `\DeclareSectionCommandsauter`.

La même chose arrive avant d'appeler `\chapterheadendvskip` avec la valeur résultant de `\DeclareSectionCommand`, et d'appeler `\chapterheadmidvskip` avec la valeur du paramètre `\DeclareSectionCommand`.

`\chapterheadstartvskip`, `\chapterheadmidvskip` ou `\chapterheadendvskip` sont redéfinis et les distances configurées avec `\RedeclareSectionCommand`, vous devriez donc également utiliser la nouvelle définition `\@tempskipa`.

Étant donné que les paramètres par défaut pour les distances de `\part` ne dépendent pas de l'option `headings`, les instructions associées à cette option ne sont pas redéfinies. Vos définitions originales respectent `scrbook` et `scrreprt` :



```

\newcommand*{\partheadstartvskip}{%
\null\vskip-\baselineskip\vskip\@tempskipa
}
\newcommand*{\partheadmidvskip}{%
\par\nobreak
\vskip\@tempskipa
}
\newcommand*{\partheadendvskip}{%
\vskip\@tempskipa\newpage
}

```

et avec `scrartcl` :

```

\newcommand*{\partheadstartvskip}{%
\addvspace{\@tempskipa}%
}
\newcommand*{\partheadmidvskip}{%
\par\nobreak
}
\newcommand*{\partheadendvskip}{%
\vskip\@tempskipa
}

```

Encore une fois `\@tempskipa` est réglé avant d'utiliser les commandes au sein de `\part` en fonction des paramètres de `\DeclareSectionCommand`. Étant donné que les distances à l'intérieur et sous les rubriques peuvent être définies avec `\RedeclareSectionCommand`, il n'est pas conseillé de redéfinir les instructions décrites ici, qui devraient être réservées à des changements plus profonds non accessibles via `\RedeclareSectionCommand`. Sur [Koh], on trouve à ce propos les redéfinitions de `\chapterheadstartvskip` et `\chapterheadendvskip` au-dessus et en dessous du titre de chapitre.

```

\chapterlineswithprefixformat{niveau}{nombre}{texte}
\chapterlinesformat{niveau}{nombre}{texte}

```

Ces deux états sont utilisés par les rubriques du chapitre de style en fonction de l'option `chapterprefix`. Ici, `\chapterlineswithprefixformat` active la sortie avant, alors que `\chapterlinesformat` détermine quand elle n'est pas activée. Les deux arguments, nombre et texte, permettent déjà la définition des polices pour compléter chaque format et règlent les instructions de disposition des deux parties de la rubrique. Pour les rubriques non numérotées, nombre est un argument complètement vide, donc ne contient aucune instruction de

formatage. Les deux déclarations plutôt spartiates jusque-là sont prédéfinies par défaut :

```
\newcommand{\chapterlinesformat}[3]{%
\@hangfrom{#2}{#3}%
}
\newcommand{\chapterlineswithprefixformat}[3]{%
#2#3%
}
```

Vous souhaitez donc définir le style des rubriques de chapitre en jaune pour les titres sans ligne de préfixe dans le préambule du document :

```
\makeatletter
\renewcommand{\chapterlinesformat}[3]{%
\colorbox{yellow}{%
\parbox{\dimexpr\linewidth-2\fbboxrule-2\fbboxsep}{%
\@hangfrom{#2}{#3}%
}%
}%
}
\makeatother
```

et les titres avec la ligne de préfixe :

```
\renewcommand{\chapterlineswithprefixformat}[3]{%
\colorbox{yellow}{%
\parbox{\dimexpr\linewidth-2\fbboxrule-2\fbboxsep}{%
#2#3%
}%
}%
}
```

Vous remarquez après un certain temps qu'avec cette redéfinition, les positions sont à nouveau justifiées. La raison se trouve dans l'argument de `\parbox`. Pour corriger cela, ajouter la commande `\raggedchapter` (voir la section 3.16), qui est automatiquement appelée dans les définitions avant `\chapterlinesformat` et `\chapterlineswithprefixformat` :

```
\makeatletter
\renewcommand{\chapterlinesformat}[3]{%
\colorbox{yellow}{%
\parbox{\dimexpr\linewidth-2\fbboxrule-2\fbboxsep}{%
\raggedchapter{#2}{#3}%
}%
}%
}
```

```

\raggedchapter
\@hangfrom{#2}#3%
}%
}%
}
\makeatother
\renewcommand{\chapterlineswithprefixformat}[3]{%
\colorbox{yellow}{%
\parbox{\dimexpr\linewidth-2\fbboxrule-2\fbboxsep}{%
\raggedchapter
#2#3%
}%
}%
}

```

Il est rappelé que `\makeatletter` et `\makeatother` ne sont pas à utiliser dans le préambule du document. Dans une classe `Wrapper` séparée ou dans un `paq`, ils doivent être éliminés. Ils ne sont nécessaires qu'avec `\@hangfrom` dans la définition de `\chapterlinesformat`.

Comme le montre l'exemple, l'utilisateur est responsable des redéfinitions des commandes pour certaines choses elles-mêmes. En plus de recevoir une partie du pré-alignement dans la rubrique, aucun exemple de pagination des paragraphes ou des intervalles ne peut être inséré. L'exemple représenté à cet égard est sans problème. Non seulement les deux boîtes ne permettent pas, de toute façon, la pagination, mais `KOMA-Script` a également changé `\interlinepenalty` dans le cadre du texte de sorte qu'un saut de page soit impossible. Simultanément, le texte se termine toujours avec un paragraphe interne.

`\@par`

La commande `\raggedchapter`, ne fait pas partie du texte sinon l'utilisation de `\MakeUppercase` au sein de la redéfinition de `\chapterlinesformat` ou de `\chapterlineswithprefixformat` serait entravée.

Noter que les règles typographiques fixées pour les polices communes exigent le blocage et une compensation que l'instruction  $\text{\LaTeX}$  `\MakeUppercase` ne réalise pas. Le premier argument, `niveau`, n'est pas utilisé par les instructions dans le réglage par défaut et n'est pas non plus dans l'échantillon nécessaire. Seulement si l'utilisateur définit une série d'énoncés dans le chapitre de style et que vous voulez faire une distinction de la définition selon les instructions, cela peut être fait par `niveau`. Il existe le `niveau` du nom de l'instruction dérivé du

niveau hiérarchique. Dans `\chapter`, `\chapter*`, `\addchap` et `\addchapter*` le niveau uniforme est donc `chapter`.

```
\sectionlinesformat{niveau}{collection}{nombre}{texte}
\sectioncatchphraseformat{niveau}{collection}{nombre}{texte}
```

Ces deux états sont des rubriques de la `section` du style utilisé selon que les titres génèrent une *Spitzmarke*<sup>1</sup> ou une position autonome.

Les rubriques détachées sont ainsi délivrées par `\sectionlinesformat`, tandis que `\sectioncatchphraseformat` est responsable des *Spitzmarke*.

Dans les deux cas, la collecte spécifie la valeur de l'espace horizontal entre le titre et la zone de texte. Entrer une valeur négative devrait rendre possible de mettre ce titre dans la marge de gauche. Les deux arguments, nombre et texte définissent les polices de chaque format pour, en fin de compte, régler les instructions de disposition des deux parties de la rubrique.

Pour les rubriques non numérotées, le nombre est un argument vide, sans aucune instruction de formatage et les deux déclarations sont prédéfinies, comme :

```
\newcommand{\sectionlinesformat}[4]{%
  \@hangfrom{\hskip #2#3}{#4}%
}
\newcommand{\sectioncatchphraseformat}[4]{%
  \hskip #2#3#4%
}
```

Dans la redéfinition de l'une des deux directions, à son tour, l'utilisateur lui-même est responsable de sauts de page pour empêcher la sortie de l'intérieur. KOMA-Script définit, en conséquence, uniquement `\interlinepenalty`.

Comme dans l'exemple des têtes de chapitre dans les rubriques individuelles à plusieurs niveaux, les sections seront maintenant identifiées avec une couleur, et les rubriques de niveaux inférieurs ne devraient pas être modifiées :

```
\makeatletter
\renewcommand{\sectionlinesformat}[4]{%
  \@tempwafalse
  \ifstr{#1}{section}{%
    \hspace*{#2}%
    \colorbox{yellow}{%

```

---

1. Mot utilisé en tant qu'information d'introduction, appelé au début d'un message ou d'un communiqué de presse et par extension dans l'usage présent, marque de la première ligne d'un paragraphe (un substitut à l'intitulé identifiant).

```

\parbox{\dimexpr\linewidth-2\fbboxrule-2\fbboxsep-#2}{%
\raggedsection
\@hangfrom{#3}{#4}%
}%
}%
}{%
\@hangfrom{\hskip #2#3}{#4}%
}%
}
\makeatother

```

Le code indiquant la zone du dispositif d'alimentation n'est pas coloré dans le cas d'un retrait du titre. Si le titre est placé dans la marge de gauche, la zone de bordure est aussi mise en évidence en couleur. Ce comportement peut changer en déplaçant la commande `\hspace*` dans le répertoire `\colorbox`.

Il est rappelé que `\makeatletter` et `\makeatother` ne s'utilisent pas dans le préambule du document et doivent être éliminés du paq ou d'une classe `Wrapper` séparée. Ils ne sont nécessaires que pour `\@hangfrom` dans la définition de `\sectionlinesformat` premier argument, niveau, mais ne sont pas requis par les instructions de la valeur par défaut. Il peut être utilisé par différents niveaux de détail, dans le style commun, pour distinguer la section.

```

\SecDef{déclaration étoilée}{déclaration normale}
\scr@startsection{nom}{niveau}{famille}{espace avant}{espace après} {déclaration de style}[forme abrégée]{titre}
\scr@startsection{nom}{niveau}{famille}{espace avant}{espace après} {déclaration de style}*{titre}
\At@startsection{code}
\Before@ssect{code}
\Before@sect{code}

```

Dans la description des définitions de commandes, [section 3.16](#), `KOMA-Script` détaille les caractéristiques concernant l'argument optionnel des commandes de sectionnement à propos des options avancées indispensables pour modifier les instructions « core » de `LATEX` comprenant `\secdef` et `\@startsection`. Pour en connaître la signification des paramètres, se référer aux instructions pour « core » de `LATEX` [[Bra+05](#)].

Ces déclarations redéfinissent souvent complètement les paqs et les fonctions de `KOMA-Script` peuvent en être sérieusement impactées, les dites déclarations `LATEX-core` ne sont pas simplement redéfinies et des instructions alternatives

supplémentaires `\SecDef` et `\scr@startsection` peuvent être utilisées par les auteurs de paqs, avec à la clé une offre de fonctionnalités étendues. Toutefois, ces deux déclarations ne doivent pas être redéfinies car susceptibles d’être modifiées à tout moment et les fonctionnalités de KOMA-Script en seraient, à nouveau, affectées.

KOMA-Script utilise en interne `\SecDef` et `\scr@startsection` en place de `\secdef` et `\@startsection` par exemple dans la division des commandes qui sont définies avec `\DeclareSectionCommand`. Une redéfinition ultérieure de `\secdef` et `\@startsection` n’a donc pas d’impact sur ces commandes.

En remplacement de la redéfinition des commandes, KOMA-Script peut insérer du code supplémentaire dans l’exécution de ses propres extensions.

Le code d’appel vers les commandes `\At@startsection`, `\Before@sect` et `\Before@@sect` est collecté séparément pour des instructions individuelle et ne vise pas à supprimer le code en cas de nouvelle insertion.

Le code de `\At@startsection` commence dans `\scr@startsection` après la `section` d’évaluation de l’espace des paramètres « avant » et le calcul associé de la résultante `\@tempskipa` toujours appliqué avant d’insérer la distance.

Il est donc possible de changer, sur cette distance, `\@tempskipa`. Le code `\Before@sect` ou `\Before@ssect` démarre immédiatement avant d’appeler, lors de l’exécution de `\scr@startsection`, respectivement `\@sect` ou `\@ssect`.

À ce moment, le cas échéant, la distance verticale est insérée avant l’intertitre via `\advspace`.

Les instructions `\At@startsection`, `\Before@sect` et `\Before@ssect` sont conçues pour les auteurs de paqs et approuvées pour cet usage.

#### `\appendixmore`

Dans les classe KOMA-Script, la commande `\appendix` intègre une fonction spéciale : si `\appendixmore` est défini, la déclaration `\appendix` est exécutée. Les classes internes de KOMA-Script, `scrbook` et `scrreprt` permettent de réaliser l’option de mise en page `appendixprefix` utilisée (voir la `section 3.16`). Vous devrez vous le rappeler pour définir ou redéfinir la macro `\appendixmore`.

Si cette option est utilisée, vous aurez avec `\renewcommand{\appendixmore}{...}` un message d’erreur afin de vous empêcher de changer, sans le savoir, les options en vigueur.

Si vous ne voulez pas, lors de l'utilisation des classes `scrbook` ou `scrreprt`, que la partie principale du chapitre soit dotée d'une ligne de préfixe (voir option de mise en page dans la section 3.16) et pour garder une cohérence, pas de ligne non plus dans l'annexe et le mot « annexe » devant la marque du chapitre dans la langue appropriée, vous décidez de ne pas utiliser l'option de mise en page `appendixprefix` mais de définir le préambule dans le document, ce qui est également valable pour le titre de la colonne :

```
\newcommand*\appendixmore>{%
  \renewcommand*\chapterformat{%
    \appendixname~\thechapter\autodot\enskip}%
\renewcommand*\chaptermarkformat{%
  \appendixname~\thechapter\autodot\enskip}}
```

Si par la suite vous utilisez l'option `appendixprefix` et que vous obtenez, en raison de l'instruction `\appendixmore` déjà définie, un message d'erreur, il doit éviter que la définition comportant des paramètres inaperçus n'écrase ce que vous avez mis dans votre option. Obtenir un comportement similaire de l'annexe pour la classe `scrartcl` est également possible. L'exemple de la ligne suivante est à ajouter dans le préambule de votre document.

```
\newcommand*\appendixmore>{%
\renewcommand*\sectionformat{%
\appendixname~\thesection\autodot\enskip}%
\renewcommand*\sectionmarkformat{%
\appendixname~\thesection\autodot\enskip}}
```

Une explication des instructions indéfinies de cet exemple dans la section 3.16.

```
\newbibstyle[style parent]{nom}{instruction}
\newblock
\@openbib@code
\bib@beginhook
\bib@endhook
```

Même les classes `standard` répartissent les entrées dans la bibliographie par la commande `\newblock` avec un résultat précis. Si l'option `openbib` est utilisée, les commandes `\@openbib@code` et `\newblock` se redéfinissent à la fin de la classe par défaut. Si l'option `openbib` est utilisée, elles le sont à la fin de la classe `standard`. Dans les classes `standard`, la commande `\@openbibcode` est au début de la liste de la bibliographie - plus précisément dans la définition

des paramètres de cette liste - en cours d'exécution. On peut supposer que de nombreux paqs, qui redéfinissent la bibliographie, exécutent cette déclaration en conséquence.

Dans les classes KOMA-Script, quelque chose de semblable se produit. Ce n'est pas `\@openbib@code` qui est redéfini à la fin de la classe, mais le style `openstyle|seebibliography` avec `\newbibstyle`. Les déclarations ainsi données dans la mise en œuvre ont été ajoutées, y compris la redéfinition nécessaire de `\@openbib@code` et `\newblock`. Maintenant en utilisant l'option `bibliography=openstyle`, le style sélectionné de bibliographie est choisi de sorte que les instructions sont exécutées immédiatement, et donc `\@openbib@code` et `\newblock` redéfinis.

Outre `\@openbib@code` et `\newblock`, les instructions `\bib@beginhook` et `\bib@endhook` peuvent également être redéfinies.

La commande `\@bibbeginhook` est exécutée immédiatement après le titre et le préambule de la bibliographie, mais avant la liste des entrées en cours. La commande `\@bibendhook` est exécutée immédiatement après cette liste à la fin de la bibliographie. En cas d'interruption de `\BreakBibliography` (voir l'article 23.3), ces énoncés sont également au début et à la fin de chaque partie, donc exécutés immédiatement avant par `\BreakBibliography`.

Les instructions utilisées pour un nouveau style de bibliographie, `\newblock`, `\@openbib@code`, `\bib@beginhook` et `\bib@endhook` sont initialement vides par défaut.

Le style, puis les instructions enregistrées dans la définition du nouveau style sont exécutés. Il en résulte que chacun des quatre énoncés contenus dans l'instruction est défini avec `\renewcommand` et n'a pas besoin de `\newcommand`. L'utilisateur définit plus d'instructions pour l'exécution, avec les commandes `\AtEndBibliography` et `\AfterBibliographyPreamble` par le préambule et à la fin de la bibliographie. Avec `\AfterBibliographyPreamble` il établit au début de la bibliographie une fois des instructions selon `\bib@beginhook` qui sont uniquement exécutée avec `\AtEndBibliography` à la fin de la bibliographie, avant `\bib@endhook`.

L'utilisation du paq `multicol` (voir [Mit11]) pourrait être, par exemple, un style littéraire pour la définition d'une liste de références à deux colonnes :

```
\newbibstyle{twocolumstyle}{%
\renewcommand*{\bib@beginhook}{\begin{multicols}{2}}%
\renewcommand*{\bib@endhook}{\end{multicols}}}%
```



Dans une variante `open`, vous pouvez utiliser ici les possibilités d'hérité et spécifier, lorsque vous créez un style « parent » :

```
\newbibstyle{twocolumopenstyle}[openstyle]{%
\renewcommand*{\bib@beginhook}{\begin{multicols}{2}}%
\renewcommand*{\bib@endhook}{\end{multicols}}}%
```

La sélection de ce nouveau style se fait aisément dans l'option bibliographie.

## 20.4 Options et instructions obsolètes

Plus d'information figurent, à ce propos, dans le livre `KOMA-Script` [Koh14a].



# Chapitre 21

## Informations supplémentaires

Dans ce chapitre, vous trouverez des informations supplémentaires sur la classe KOMA-Script `scr1tr2` et le paq `scrletter`.

Certaines parties de ce chapitre extraites de KOMA-Script livre [KM12] sont réservées. Cela ne devrait pas être un problème car l'utilisateur qui veut juste utiliser la classe n'a pas besoin, normalement, de ces indications. Certaines sont destinées aux utilisateurs qui ne remplissent plus les options prédéfinies, telles que les premières offres de `section` avec les pseudo-longueurs qui déterminent l'entête et doivent être réglées différemment pour s'adapter à une disposition personnalisée. Elles fournissent également des informations d'amélioration des possibilités de compatibilité avec les versions KOMA-Script créées à l'aide de la classe `scrletter` obsolète. Elles expliquent aussi comment modifier un document d'une ancienne classe pour l'utiliser avec la classe `scr1tr2` actuelle.

### 21.1 Les pseudo-longueurs

Ce chapitre est destiné aux utilisateurs expérimentés -  $\TeX$  fonctionne avec un nombre fixe de registres – registres de jetons, pour les boîtes, les compteurs, les distances (en anglais : *skip*) et tailles (en anglais : *dimension*). Il en existe 256 différents. Pour les longueurs de  $\LaTeX$  qui sont demandées avec `\newlength`, les registres sont utilisés à distance. Si tous ces registres sont utilisés, on ne peut définir des longueurs supplémentaires.

La classe `sclttr2` en consomme normalement plus de vingt, seulement pour la première page.

$\LaTeX$  lui-même en utilise déjà quarante et, pour le paq de `typearea`, un quart de ces précieux registres serait consommé.

Pour cette raison, les longueurs spécifiques aux lettres dans `sclttr2` ne sont tout simplement pas stockées dans les longueurs, mais dans les macros comme pseudo-longueurs. L'inconvénient de cette approche est que vous ne pouvez pas calculer simplement avec ces macros comme avec des longueurs réelles. Rappelez-vous que les pseudo-longueurs sont mises en œuvre en interne comme macros, mais que seul leur nom est spécifié dans les instructions d'utilisation. Cela peut être écrit comme les noms des compteurs  $\LaTeX$ , sans `controble`<sup>1</sup>, contrairement aux macros ou aux longueurs réelles. Celui qui prétend que dans les recommandations pour  $\LaTeX$  et `KOMA-Script` ne figurent plus les scripts utiles au traitement du problème d'installation limitée pour  $\varepsilon\text{-}\TeX$  a raison. Cependant, cette décision a été prise un peu trop tard pour `sclttr2`.

Une liste de toutes les définitions et pseudo-longueurs utilisées par `sclttr2` est présentée dans la table 21.1. Il est indiqué où des explications plus détaillées peuvent être consultées sur les pseudo-longueurs dans les paragraphes suivants, voir la figure 21.1 page 579 qui montre une représentation schématique des lacunes les plus importantes dans l'entête de lettre. Voici en outre, indiquées à côté des pseudo-longueurs pour intervalles variables en caractères lumineux, les longueurs utilisées pour certains intervalles pré-programmés. Cependant, pour des raisons de clarté, certaines pseudo-longueurs moins souvent requises ont été volontairement omises dans l'illustration.

TABLE 21.1 – Pseudo-longueurs utilisées par la classe `sclttr2`

---

<code>backaddrheight</code>	adresse de retour sur le dessus de la zone d'adresse ( <a href="#">section 22.1.3</a> )
<code>bfoldmarklength</code>	longueur de la marque inférieure de pliage horizontal (de la <a href="#">section 22.1.1</a> )
<code>bfoldmarkvpos</code>	distance de la marque inférieure de pliure horizontale au bord supérieur du papier ( <a href="#">section 22.1.1</a> )

---

1. Voir la note de la page xxv.

---

Pseudo-longueurs utilisées par la classe `scrlttr2` (suite)

---

**firstfootpos**

distance du pied de la lettre au bord gauche du papier ; les valeurs supérieures à la largeur ou plus petites que la largeur négative du papier sont traitées séparément (section 22.1.8)

**firstfootvpos**

distance du pied de la lettre au bord supérieur du papier (section 22.1.8)

**firstfootwidht**

largeur du pied de la lettre (article 22.1.8)

**firsttheadpos**

distance de l'entête au bord gauche du papier ; les valeurs supérieures à la largeur ou plus petites que la largeur négative du papier sont traitées séparément (section 22.1.2)

**firsttheadvpos**

distance de l'entête au bord supérieur du papier (section 22.1.2)

**firsttheadwidth**

largeur de l'entête (section 22.1.2)

**Foldmarkhpos**

distance entre la marque horizontale de pliage et le bord gauche du papier (section 22.1.1)

**foldmarkvpos**

espace de la marque de pliage vertical au bord supérieur du papier (section 22.1.1)

**fromrulethickness**

épaisseur d'une ligne facultative dans l'entête (section 22.1.2)

**fromrulewidth**

longueur d'une ligne facultative dans l'entête (section 22.1.2)

**lfoldmarkhpos**

espace de la marque du pliage vertical au bord gauche du papier (section 22.1.1)

**lfoldmarklength**

longueur de la marque du pliage vertical (section 22.1.1)

**locheight**

hauteur de l'extension de l'expéditeur, si la valeur n'est pas 0 ; avec 0, `toaddrheight` sera utilisé à la place (section 22.1.4)

**lochpos**

distance de l'extension de l'expéditeur au bord droit du papier si la valeur est

---

---

Pseudo-longueurs utilisées par la classe `scrlettr2` (suite)

---

une distance positive ou distance de l'extension de l'expéditeur au bord gauche du papier si la valeur est négative ; si la valeur est à 0, la valeur de `toaddrhvpos` sera utilisée à la place ([section 22.1.4](#))

-----  
`locvpos`

distance de l'extension de l'expéditeur au bord supérieur du papier, si la valeur n'est pas 0 ; si la valeur est 0, `toaddrvpos` est utilisé à la place ([section 22.1.4](#))

-----  
`locwidth`

largeur du champ pour l'extension de l'expéditeur. Avec une valeur de 0, la largeur est automatiquement basée sur l'option décrite à la [section 4.10](#) et calculée par `locfield` ([article 22.1. 4](#))

-----  
`mfoldmarklength`

longueur de la marque horizontale de pliage central ([article 22.1.1](#))

-----  
`mfoldmarkvpos`

distance du pli horizontal de la marque centrale au bord supérieur du papier ([section 22.1.1](#))

-----  
`pfoldmarklength`

longueur de la marque du poinçon ([section 22.1.1](#))

-----  
`refaftervskip`

distance verticale selon l'empage ([article 22.1.5](#))

-----  
`refhpos`

positionnement de la ligne à partir du bord gauche du papier, une valeur de 0 déclarée sera automatiquement centrée par rapport à la largeur du papier

-----  
`refvpos`

positionnement de la ligne à partir du bord supérieur du papier ([section 22.1.5](#))

-----  
`refwidth`

largeur de la ligne ([section 22.1.5](#))

-----  
`sigbeforevskip`

distance verticale entre salutation et signature ([article 22.1. 7](#))

-----  
`sigindent`

position de la signature par rapport au corps du texte ([section 22.1.7](#))

-----  
`specialmailindent`

tiret gauche de service dans le champ d'adresse ([section 22.1.3](#))

-----  
`specialmailrightindent`

tiret droit de service dans le champ d'adresse ([section 22.1. 3](#))

-----  
`subjectaftervskip`

distance verticale après le sujet ([section 22.1.6](#))

---

---

Pseudo-longueurs utilisées par la classe `scrlltr2` (suite)

---

`subjectbeforevskip`

distance verticale supplémentaire à partir de l'objet (section 22.1.6)

`subjectvpos`distance du sujet à partir du bord supérieur du papier, la valeur de 0 est recon-  
nue selon l'option de l'objet (section 22.1.6)`tfoldmarklength`

longueur des marques de pliage horizontal supérieures (section 22.1.1)

`tfoldmarkvpos`distance entre les marques de pliage horizontal supérieures au bord supérieur  
du papier (section 22.1.1)`toaddrheight`

hauteur de la zone d'adresse (section 22.1.3)

`toaddrhpos`si la valeur est positive, distance du champ d'adresse à partir du bord gauche  
du papier, ou décalage du champ d'adresse à partir du bord droit du papier, si  
la valeur est négative (section 22.1.3)`toaddrindent`

tiret à gauche et à droite de l'adresse dans le champ d'adresse (section 22.1.3)

`toaddrvpos`distance de la zone d'adresse depuis le bord supérieur du papier (section  
22.1.3)`toaddrwidth`largeur de la zone d'adresse (section 22.1.3)

---

`\@newplength{nom}`

Grâce à cette commande, une nouvelle pseudo-longueur est définie, qui peut être clairement identifiée par son nom. Tenter de redéfinir une pseudo-longueur existante provoque un message d'erreur. L'utilisateur normalement ne définit pas ses propres pseudo-longueurs, il n'existe donc aucun manuel d'utilisation pour cette commande.

`\@setlength[facteur]{pseudo-longueur}{valeur}``\@addtoplength[facteur]{pseudo-longueur}{valeur}`

Avec la macro `\@setlength` une pseudo-longueur peut être un multiple d'une valeur ajoutée rapportée. Le facteur est passé comme un argument optionnel (voir aussi `\@setlengthtoplength`, section 4.2).

Avec `\@addtolengthlength` vous pouvez ajouter une pseudo-longueur du multiple d'une valeur. Le facteur est donné comme argument optionnel ici aussi. On peut attribuer ou ajouter à une pseudo-longueur un multiple d'une autre pseudo-longueur en utilisant une valeur de la commande `\useplength` (voir la section 4.2). Pour soustraire une pseudo-longueur de la valeur d'une autre pseudo-longueur, on utilise simultanément comme facteur un signe moins, `-1` ou un autre *facteur négatif*.

Puisque l'utilisateur ne doit généralement changer aucune pseudo-longueur, il n'existe pas de mode d'emploi de ces commandes qui peuvent être utilisées, par exemple dans un fichier `.ico`, mais pas dans le document.

### 21.1.1 Marques de pliage

Les pliures ou marques de pliage sont de petites barres horizontales et/ou verticales sur le côté gauche ou le haut de la feuille. `KOMA-Script` supporte actuellement trois marques horizontales et une marque verticale configurables. À cet effet, un trou (ou une marque) centré horizontalement sur un côté est toujours pris en charge, et ne peut être décalé dans le sens vertical.

```
\@setplength{tfoldmarkvpos}{longueur}  
\@setplength{mfoldmarkvpos}{longueur}  
\@setplength{bfoldmarkvpos}{longueur}
```

La classe lettre `scr1ttr2` reconnaît un total de trois verticales configurables dans le placement de pliages. La position de la marque de pliage sur le bord supérieur du papier est déterminée par la pseudo-longueur `tfoldmarkvpos`, la pliure au centre est gérée par la pseudo-longueur `mfoldmarkvpos`, alors que la marque de pliage inférieure l'est par `bfoldmarkvpos`. La marque du poinçon en milieu de page représente une autre marque horizontale, toujours placée au centre vertical de la page. Les plis supérieur et inférieur ne marquent pas le tiers du papier exactement mais le papier doit être plié avec leur aide, de sorte que le champ d'adresse puisse être vu dans une enveloppe à fenêtre. Les réglages sont donc choisis différemment dans les fichiers `.ico` prédéfinis. Une anomalie est présente avec `DINmtext` : un format d'enveloppe C6/5 (c'est-à-dire connu comme « C6 long ») est supposé. Les lettres créées avec cette option ne sont pas adaptées pour une enveloppe C5, ni pour le format des enveloppes C4. La marque centrale n'est pas indispensable pour les lettres occidentales, mais au Japon, il existe tellement de formats d'enveloppe que cette marque de pliage est nécessaire (voir les fichiers `.ico` japonais). À ce stade, il convient



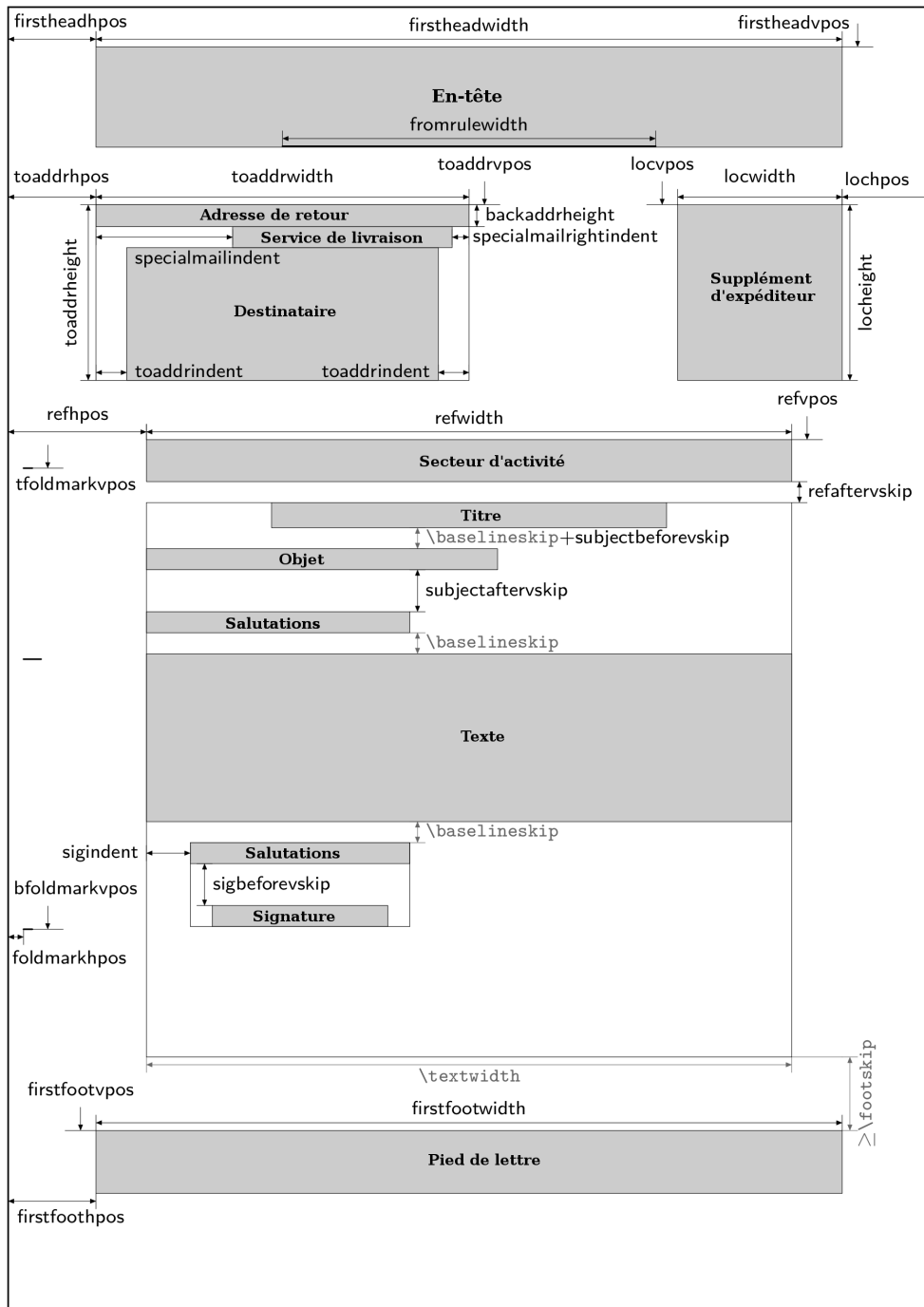


FIGURE 21.1 – Représentation schématique des pseudo-longueurs les plus importantes pour entête

de souligner que les termes le plus élevé, moyen et le plus bas ne sont que des conventions de langage. En fait, rien n'établit que `tfoldmarkvpos` est inférieur à `mfoldmarkvpos`, qui doit être lui-même inférieur à `bfoldmarkvpos`. Si l'une des pseudo-longueurs a une valeur de zéro (null), alors la pliure correspondante ne sera pas réglée même si les options des marques de pliage (voir la section 4.10) sont explicitement activées.

```
\@setlength{tfoldmarklength}{longueur}  
\@setlength{mfoldmarklength}{longueur}  
\@setlength{bfoldmarklength}{longueur}  
\@setlength{pfoldmarklength}{longueur}
```

Ces pseudo-longueurs déterminent les longueurs des marques horizontales de pliage. Un comportement particulier existe. Si la longueur donnée est nulle, les marques `tfoldmarklength`, `mfoldmarklength` et `bfoldmarklength` des trois pseudo-longueurs utilisent dans la position verticale, une longueur de 2 mm, la longueur de la marque du poinçon `pfoldmarklength` est fixée à 4 mm.

```
\@setlength{foldmarkhpos}{longueur}
```

Cette pseudo-longueur donne la distance de toutes les marques du pliage horizontal au bord gauche du papier, habituellement 3,5 mm. Vous pouvez en modifier la valeur y compris dans votre propre fichier `.ico` si vous utilisez une imprimante avec une marge de gauche plus large non imprimable. Fixer les marques de pliage dépend de l'option `foldmarks` (voir section 4.10).

```
\@setlength{lfoldmarkhpos}{longueur}
```

Outre les marques du pliage horizontal existe une marque verticale dont la distance à partir du bord gauche du papier est déterminée par les pseudo-longueurs `lfoldmarkhpos`. Nécessaire dans le cas d'enveloppes de lettres pour certains *Chou* ou *You* japonais utilisées avec le papier A4, elles peuvent être aussi utiles pour les enveloppes au format C6.

```
\@setlength{lfoldmarklength}{longueur}
```

La pseudo-longueur `lfoldmarklength` détermine la longueur de la marque de pliure verticale, avec, encore une fois, la particularité que, pour une longueur spécifiée de zéro, 4 mm seront utilisés.

```
\@setlength{lfoldmarkhpos}{longueur}
```

Donne la distance des marques du pliage vertical à partir du bord supérieur de la feuille, habituellement 3,5 mm. Vous pouvez en modifier la valeur y compris dans votre fichier `.ico` si vous utilisez une imprimante avec une marge de

gauche plus large non imprimable. Fixer toutes les marques du pliage dépend de l'option `foldmarks` (voir section 4.10). À présent, une seule pliure verticale existe, désignée comme marque du pliage vertical gauche.

```
\@setlength{foldmarkthickness}{longueur}
```

Cette pseudo-longueur donne l'épaisseur de toutes les marques du pliage. La valeur par défaut est de la finesse d'un cheveu, 0,2 pt.

## 21.1.2 Entête de lettre

L'entête comprend, au-dessus de l'adresse, les informations relatives à l'expéditeur et, pour `scr1ltr2`, est indépendant de l'instruction `\opening` de la page de style, en position absolue. La première page d'une lettre, donc le côté avec l'entête, est en fait fixée avec le style de page vide.

```
\@setlength{firstheadvpos}{longueur}
```

La pseudo-longueur `firstheadvpos` indique la distance entre l'entête et le bord haut de la feuille de papier. La valeur est définie différemment dans les fichiers `.ico` prédéfinis. Une valeur typique est de 8 mm.

```
\@setlength{firstheadhpos}{longueur}
```

La pseudo-longueur `firstheadhpos` donne une valeur positive à la distance de l'entête au bord gauche du papier. Si la valeur est supérieure ou égale à la largeur du papier `\paperwidth`, l'entête est centré horizontalement. Une valeur négative indique la distance de l'entête à partir du bord droit du papier. Si la valeur est inférieure ou égale à la largeur négative du papier, l'entête est placé au ras du bord gauche de la zone du style. La valeur par défaut est une valeur de `\maxdimen`, c'est-à-dire la plus grande valeur possible pour une longueur. Le résultat est un centrage horizontal.

```
\@setlength{firstheadwidth}{longueur}
```

La pseudo-longueur `firstheadwidth` spécifie la largeur de l'entête avec sa valeur définie différemment dans les fichiers `.ico` prédéfinis. Si elle dépend généralement de la largeur du papier et de la distance horizontale de l'adresse du destinataire à partir du bord gauche du papier, elle correspond à la largeur du type `KOMAold` fixée à 170 mm pour les NF.

```
\@setlength{fromrulethickness}{longueur}  
\@setlength{fromrulewidth}{longueur}
```

Comme mentionné à la section 4.10, une ligne peut être réglée dans les entêtes prédéfinis, avec ou sans l'expéditeur, à l'aide de l'option `fromrule`. La pseudo-longueur `fromrulewidth` a une longueur de 0 qui permet de déterminer la longueur de la ligne automatiquement. C'est la valeur par défaut dans les fichiers `.ico` prédéfinis. La valeur peut être réglée avec `\@setlength` dans ses propres fichiers `.ico` mais aussi définie différemment. L'épaisseur de la ligne par défaut est de 0,4 pt.

### 21.1.3 Adresse

L'adresse fait référence aux nom et adresse du destinataire avec une première extension qui peut être considérée comme utilisée par exemple dans l'écriture des lettres ou des informations. Avec des enveloppes à fenêtre, l'adresse de retour sera comptée dans l'adresse et affichée dans la fenêtre d'adresse. L'adresse est placée immédiatement après l'entête.

```
\@setlength{toaddrvpos}{longueur}  
\@setlength{toaddrhpos}{longueur}
```

Ces pseudo-longueurs indiquent la distance de la fenêtre d'adresse à partir du haut et du bord gauche du papier. Elles sont réglées d'une façon différente dans les fichiers `.ico` prédéfinis. Il existe également une fonction spéciale pour `toaddrhpos`. Si la valeur est négative, la valeur sera la distance entre le champ d'adresse jusqu'au bord droit de la feuille. Vous pouvez trouver cet exemple dans SN ou NF. La plus petite valeur est `toaddrvpos` avec `DINmtext`. Il peut se produire que l'entête domine dans la fenêtre d'adresse dont le réglage dépend d'une façon générale de l'option `addrfield` (voir section 4.10).

```
\@setlength{toaddrheight}{longueur}
```

Cette pseudo-longueur spécifie la hauteur de la zone d'adresse y compris la méthode d'expédition. En son absence, l'adresse est centrée verticalement à l'intérieur du champ. Si une méthode d'envoi est spécifiée, l'adresse est centrée verticalement au-dessous de l'expéditeur dans le champ d'adresse.

```
\@setlength{toaddrwidth}{longueur}
```

Cette pseudo-longueur définit la largeur de la fenêtre d'adresse. Elle est réglée différemment dans les fichiers `.ico` prédéfinis selon des normes différentes. Des valeurs typiques sont comprises entre 70 mm et 100 mm.

Exemple : votre imprimante dispose d'une marge gauche ou droite très large de 15 mm non-imprimable. Dans ce cas, lorsque vous utilisez l'option SN, l'entête, les extensions de l'expéditeur et l'adresse ne peuvent pas être complètement imprimés<sup>2</sup>. Ainsi, vous créez un nouveau fichier lco avec le contenu suivant :

```
\ProvidesFile{SNmmarg.lco}
      [2015/03/04 v0.1 ma marge lco perso]
\LoadLetterOption{SN}
\@addtoplength{toaddrwidth}{\%
      \useplength{toaddrhpos}}
\@setlength{toaddrhpos}{-15mm}
\@addtoplength{toaddrwidth}{\%
      \useplength{toaddrhpos}}
\endinput
```

Jusqu'à ce que vous obteniez une imprimante avec de petites marges, vous utiliserez SNmmarg au lieu de SN.toaddrindent.

```
\@setplength{toaddrindent}{longueur}
```

Parfois, vous souhaitez que l'adresse ne commence pas sur le bord gauche et s'étende vers le bord droit avec un peu de retrait. La valeur de ce dispositif d'alimentation peut être réglée sur la pseudo-longueur toaddrindent. Par défaut, elle est de 0 pt et dans chacun des paramètres addrfield=PP, addrfield=image et addrfield=backgroundimage est remplacée par un aménagement de 8 mm que l'on peut négliger avec 1 sp<sup>3</sup>. En outre toaddrindent est utilisé dans les paramètres par addrfield pour fixer la distance à la droite de la fenêtre d'adresse.

```
\@setplength{backaddrheight}{longueur}
```

Avec des enveloppes à fenêtre, l'expéditeur est en principe affiché dans une petite police sur une ligne de l'adresse du destinataire. Cette spécification est appelée l'adresse de retour, car visible dans la fenêtre d'adresse et la poste

2. Un petit rappel : entête est masculin.

3. J'imagine qu'il s'agit, dans la typographie allemande, d'un raccourci désignant l'espacement entre les lettres (Sperren) obtenu, avec les logiciels usuels de traitement de texte, par une commande du sous-menu caractère ou police du menu (un espacement étendu de 1 point suffit). L'utilisation de ce procédé, de plus en plus rare, est liée à l'usage des caractères gothiques avec lesquels une bonne partie des ouvrages étaient imprimés jusqu'au milieu du XXe siècle et qui ne permettaient pas d'écrire tout un mot en *italique* ou en CAPITALES, mais qui a été aussi utilisé pour des textes en caractères romains.

l'utilise pour le retour à l'expéditeur des lettres non distribuables. La pseudo longueur `backaddrheight` fixe la hauteur disponible dans la fenêtre d'adresse pour l'adresse de retour. La valeur est définie à 5 mm dans les fichiers prédéfinis `.ico`. Si l'adresse de retour n'est pas définie, elle est déterminée par l'utilisateur à l'aide des options `addrfield` et `backaddress` (voir section 4.10).

```
\@setlength{specialmailindent}{longueur}
\@setlength{specialmailrightindent}{longueur}
```

Une méthode facultative d'expédition peut encore être réglée entre adresse de retour et l'adresse du destinataire. Cette dernière est exactement définie si la variable a un contenu email spécial. L'orientation est définie en utilisant les pseudo-longueurs `specialmailindent` et `specialmailrightindent`. Celles-ci indiquent le retrait de la ligne à gauche et à droite. Dans les fichiers `.ico` prédéfinis, `specialmailindent` est fixé avec la valeur extensible `\fill`, alors que `specialmailrightindent` est réglée à 1 em, distance du bord droit de la fenêtre d'adresse.

```
\@setlength{PPheadheight}{longueur}
\@setlength{PPheadwidth}{longueur}
```

La pseudo-longueur `PPheadheight` est dans deux paramètres `addrfield=PP` et `addrfield=backgroundimage` et permet de régler la hauteur réservée pour le port payé dans l'entête au début de la zone d'adresse. La pseudo-longueur `PPheadwidth` est utilisée uniquement avec `addrfield=PP` (voir section 4.10) et renvoie la largeur du panneau gauche de l'entête du port payé avec le logo PP, le code postal et la ville. La largeur du champ droit avec le code de l'émetteur et la priorité est limitée par la largeur résiduelle. Dans `scrlttr2`, la valeur par défaut de 0 mm pour la pseudo-longueur `PPheadheight` est changée en 20,74 pt, et la valeur de la pseudo-longueur `PPheadwidth` en 42 mm.

```
\@setlength{PPdatamatrixvskip}{longueur}
```

Grâce à ces pseudo-longueur, la distance verticale entre l'entête port payé et la matrice de données dans `addrfield=PP` est définie (voir section 4.10). Dans `scrlttr2` la valeur par défaut change et passe de 0 mm à 9mm. La matrice de données est alignée à droite avec l'entête du port payé.

### 21.1.4 Les extensions de l'expéditeur

Surtout dans les lettres d'affaires, les espaces de l'entête et du pied de page ne sont souvent pas suffisantes pour inclure tous les détails de l'éditeur. Pour

incorporer plus d'informations, l'espace à côté de l'adresse est souvent utilisée. Dans ce manuel ce domaine est appelé *complément*, *supplément* ou *extension* de l'expéditeur.

```
\@setlength{locheight}{longueur}
\@setlength{lochpos}{longueur}
\@setlength{locvpos}{longueur}
\@setlength{locwidth}{longueur}
```

Les pseudo-longueurs `locwidth` et `locheight` donnent largeur et hauteur de l'extension de l'expéditeur, et `lochpos` avec `locvpos` marquent les distances de la droite au bord supérieur du papier. Ces valeurs, définies dans les fichiers `.ico`, sont prédéfinies généralement à 0 pt. Cela signifie que avec `\opening`, les valeurs sont d'abord fixées manuellement pour la largeur du papier, la largeur de la fenêtre d'adresse et la distance de cette fenêtre au bord gauche et au haut du papier. L'option `locfield` (voir [section 4.10](#)) est également prise en compte. Comme avec `toaddrhpos`, `lochpos` prend une position particulière avec des valeurs négatives. La distance à partir du bord droit de la feuille de papier est utilisée par `lochpos` comme distance avec le bord gauche du papier. La signification est exactement l'inverse de celle de `toaddrhpos` (voir [section 22.1.3](#)).

### 21.1.5 Occupation de ligne

L'empage peut être plus long qu'une ligne et n'être défini que si une des variables de l'occupation de ligne, au moins, n'est pas vide. Seuls les champs qui ne sont pas vides sont définis. Pour définir un champ apparemment vide, il faut donner les variables d'un contenu apparemment vide, comme un espace blanc ou un type `\null`. S'il est omis dans l'occupation de ligne, le contenu de la variable `date` sera émis à sa place. Des informations sur la façon dont les variables sont ajoutées ou supprimées dans les champs de référence peuvent être trouvées [section 22.2](#).

```
\@setlength{refvpos}{longueur}
```

Cette pseudo-longueur donne la distance de la ligne de référence au bord supérieur de la feuille.

Cette valeur est définie différemment dans les fichiers `.ico` prédéfinis. Les valeurs typiques sont comprises entre 80,5 mm et 98,5 mm.

`\@setlength{refwidth}{longueur}`  
`\@setlength{refhpos}{longueur}`

La pseudo-longueur `refwidth` spécifie la largeur disponible d'occupation de ligne. Sa valeur est généralement à 0 pt dans les fichiers `.ico` prédéfinis.

Cette valeur a une signification particulière.

Il n'est nullement établi qu'il n'y a pas la largeur disponible pour l'empage. Cela signifie que la largeur sera calculée au moyen de cette valeur avec `\opening`. Elle dépend du réglage de l'option `refline` (voir section 4.10). Dans le même temps, `refhpos` est aussi défini selon l'option. Avec `refline=wide` l'occupation de ligne est centrée, alors que le type de zone, avec `refline=narrow`, est aligné à gauche. `refwidth` est différent de zéro, la largeur de la ligne d'affaire n'est pas déterminée par l'option ligne de réf, `refhpos` indique la distance de l'occupation de ligne à partir du bord gauche du papier. Si cette distance est égale à zéro, la branche est orientée de manière que le rapport entre la distance de la ligne d'affaire à partir du bord gauche du papier soit dans le même rapport que la distance au bord droit de la feuille. La zone de type est centrée horizontalement sur la page, y compris l'occupation de ligne. En général, ces cas particuliers d'applications les plus courantes sont de peu d'intérêt. La règle la plus simple : soit laisser `refwidth` à zéro et définir la largeur et l'orientation de la ligne de référence de l'option `refline`, ou utiliser les deux, `refwidth` et `refhpos`, ensemble.

`\@setlength{refaftervskip}{longueur}`

Cette pseudo-longueur donne l'espace vertical qui devra être inséré après l'occupation de ligne. La valeur est définie dans les fichiers `.ico` prédéfinis. Elle a un effet direct sur la hauteur de la zone de texte de la première page. La valeur typique est comprise entre une et deux lignes.

### 21.1.6 Sujet de la lettre

Le sujet d'une lettre est réglé différemment selon les pays. Certains préfèrent le placer avant la salutation, d'autres le mettent par la suite. Certains groupes professionnels le veulent même partiellement avant le secteur d'activité.

`\@setlength{subjectvpos}{longueur}`

La valeur 0 pt de cette pseudo-longueur détermine l'option de position du sujet (voir section 4.10) et décrit comment les pseudo-longueurs `subjectbeforevskip` et `subjectaftervskip` suivantes jouent leur rôle. Pour chaque autre valeur, cet



objet est placé à la distance appropriée du bord supérieur du papier. Attention Il est recommandé de laisser dans ce cas un espace disponible suffisant pour que les chevauchements avec d'autres éléments restent improbables.

Exemple : quelques groupes professionnels préfèrent que le sujet vienne avant l'occupation de ligne.

Cela peut être fait comme suit en sélectionnant et en ajustant la position de l'occupation de ligne :

```
\ProvidesFile{lawsubj.lco}
      [2015/03/11 lawyers lco file]
      \@setlength{subjectvpos}{\useplength{refvpos}}
\@addtoplength{refvpos}{3\baselineskip}
\endinput
```

Si vous voulez que, entre le sujet et l'occupation de ligne, une rangée au moins reste encore libre, on doit accueillir un sujet de deux lignes maximum.

```
\@setlength{subjectbeforevskip}{longueur}
\@setlength{subjectaftervskip}{longueur}
```

Le sujet n'est pas tout à fait positionné et une distance supplémentaire doit être insérée avant ou après l'ouverture, ou même le sujet. La distance de l'objet touche éventuellement d'autres intervalles, tels que l'espacement automatique d'une ligne suivant le titre. Par défaut, aucun espace supplémentaire n'est donc inséré à ce point habituellement. La distance du sujet dans la classe par défaut est de deux lignes.

### 21.1.7 Salutations finales

Les salutations finales d'une lettre peuvent se composer de plusieurs parties : outre la phrase de clôture, la signature et le paraphe, ainsi qu'un commentaire avec la signature.

```
\@setlength{sigindent}{longueur}
\@setlength{sigbeforevskip}{longueur}
```

La phrase de clôture et la signature sont placées dans une boîte. La largeur de la zone est déterminée par les lignes les plus longues des salutations finales et du commentaire. La boîte est fixée par l'entrée de la pseudo-longueur `sigindent`. Dans les fichiers `.ico` prédéfinis, l'entrée est réglée sur 0 mm. Entre salutations finales et commentaire est inséré un espace vertical, défini par `sigbeforevskip`.

La valeur est répartie sur deux lignes dans les fichiers `.ico` prédéfinis et vous pouvez inclure votre signature dans cet espace.

### 21.1.8 Pied de lettre

La première page d'une lettre, la feuille de papier à lettre, ne contient pas seulement sa propre entête, l'entête de la lettre, elle contient également son propre pied qui ne dépend pas d'un style de page. Tous deux sont créés directement à partir de `\opening`.

```
\@setlength{firstfootvpos}{longueur}
```

Cette pseudo-longueur donne, dans la première page de la lettre, la distance du pied à partir du bord supérieur de la feuille. On s'assure également que la zone de non-texte ne s'étend pas dans le pied. Pour cela, la hauteur de la zone de texte sur la première page est réduite, si nécessaire, en utilisant `\enlargethispage`. En utilisant l'option `enlargefirstpage` (voir section 4.10) on peut s'assurer que la hauteur de la zone de texte, est, le cas échéant, également, à l'inverse, augmentée. Ainsi, la valeur de la longueur de la zone de texte `\footskip` pourra être réduite pour la compatibilité avec la version 2.9t (voir la section 4.4), sauf pour KOMAold et NF dans les fichiers prédéfinis `.ico` (voir le jeu de pieds en fonction du type de zone section 4.21). Dans ce cas, `enlargefirstpage` n'aura aucun effet. Depuis la version 2.9u, le pied devient une position à l'extrémité inférieure du papier. Ainsi, la quantité de la zone de type de l'entête peut alors aussi dépendre de la possibilité `enlargefirstpage`. Si le pied de la feuille de papier à lettre utilise l'option `firstfoot=false` (voir section 4.10) pour être désactivé, le paramètre `firstfootvpos` est ignoré et `\paperheight` est adopté à la place. Il reste alors une marge inférieure minimale de `\footskip`.

```
\@setlength{firstfoothpos}{longueur}
```

La pseudo-longueur `firstfoothpos` donne une valeur positive à la distance du pied de la feuille de papier à partir du bord gauche du papier. Si la valeur est supérieure ou égale à la largeur du papier `\paperwidth`, le pied est centré horizontalement sur la feuille de papier. Une valeur négative indique la distance du pied au bord droit du papier. Si la valeur est inférieure ou égale à la largeur négative du papier, le pied valide est placé avec le bord gauche justifié. La valeur par défaut est typiquement une valeur de `\maxdimen`, c'est-à-dire la plus grande valeur possible pour une longueur. Il en résulte un centrage horizontal.

```
\@setlength{firstfootwidth}{longueur}
```

Cette pseudo-longueur donne la largeur du pied de la première page de lettre. La valeur est la même dans les fichiers `.ico` prédéfinis avec `firstheadwidth`.

## 21.2 Variables pour utilisateur avancé

Outre la possibilité d'utiliser des variables prédéfinies, `srlttr2` fournit aussi des recommandations pour définir de nouvelles variables ou pour influencer leur utilisation automatique dans l'empage.

```
\newkomavar[désignation]{nom}
\newkomavar[désignation]{nom}
\removereffields
\defaultreffields
\addtoreffields{nom}
```

Une nouvelle variable est définie avec `\newkomavar`, elle est distinguée par le nom. Facultativement, un identifiant pour ce nom peut être spécifié en harmonie avec le nom afin d'accéder à une étiquette. Le terme est plutôt un complément au contenu d'une variable qui peut être une sortie similaire à ce contenu. Avec la déclaration `\addtoreffields`, le nom de la variable de l'occupation de ligne (voir la section 4.10) est ajouté à la fin de l'occupation de ligne si le contenu n'est pas vide. La version étoilée `\newkomavar*` correspond à la variante sans étoile avec un appel ultérieur de la commande `\addtoreffields`. Avec la variante étoilée, la variable est automatiquement ajoutée à la ligne de métier.

Si dans l'empage vous avez besoin d'un champ additionnel pour une extension, vous pouvez utiliser ce champ en option :

```
\newkomavar[numérotation directe]{myphone}
\addtoreffields{myphone}
```

ou le définir de manière plus concise avec :

```
\newkomavar*[numérotation directe]{myphone}
```

Dans le cas où une variable est définie pour l'occupation de ligne, il faut toujours lui donner une description. Avec la commande `\removereffields` toutes les variables peuvent être retirées de l'occupation de ligne.

Cela vaut également pour les variables de classe prédéfinies. l'empage est alors vide. Vous pouvez l'utiliser par exemple, si vous souhaitez modifier l'ordre

des variables dans l'occupation de ligne et pour rétablir l'ordre prédéfini dans l'occupation de ligne, utiliser la commande `\defaultreffields`. Néanmoins, on peut observer toutes les variables définies par l'utilisateur et supprimées de l'occupation de ligne. La date ne doit pas être ajoutée en utilisant la commande de l'occupation de ligne `\addtoeffields`. Au lieu de cela, elle fournit une date d'option si la date est indiquée sur la gauche, la droite ou n'apparaît pas du tout dans l'empage. Ces options ont également une influence sur la position de la date si aucune occupation de ligne n'est utilisée.

<pre>\usekomavar[instruction]{nom} \usekomavar*[instruction]{nom}</pre>
---

Les instructions `\usekomavar` et `\usekomavar*` sont comme les instructions dont il existe une version étoilée ou un argument optionnel non extensible. Si elles sont utilisées dans `\markboth`, `\markright` ou des instructions similaires vous ne devez insérer aucun `\protect`. Cela s'applique lorsque vous utilisez `\markleft` pour `scrpage2`. Cependant, la déclaration ne doit pas être utilisée à l'intérieur de `\MakeUppercase` et des déclarations similaires qui ont un impact direct sur leur argumentation, mais peuvent être spécifiées comme argument optionnel. Ainsi, le contenu d'une variable peut être obtenu, par exemple, en lettres capitales :

```
\usekomavar[\MakeUppercase]{nom}
```

<pre>\ifkomavareempty{nom}{true}{false} \ifkomavareempty*{nom}{true}{false}</pre>
---

Pour l'ensemble des fonctionnalités est important que le contenu de la variable soit élargi avec `\edef` quand c'est possible. Si cela se traduit par des espaces ou macros non extensibles comme avec `\relax`, le contenu doit aussi être considéré comme non vide à la condition que l'utilisation de variables n'entraîne pas de sortie. Cette commande ne peut être utilisée à l'intérieur de `\MakeUppercase` ou avec des instructions similaires. Mais elle est suffisamment robuste pour fonctionner comme argument de `\markboth` ou de `\footnote`.

## 21.3 Les différences dans les styles de `scrletter`

Dans la section 4.13 les rubriques de styles de page ont été documentées ainsi que le contenu des variables d'entête et de pied de page, `nexthead` et `nextfoot`. Depuis, cette page de style est aussi définie ailleurs dans les classes KOMA-Script, tandis que `scrletter` définit une nouvelle paire de style de page

en utilisant `scrlayer-scrpage` (voir l'article 18.2). Grâce à cet usage interne de `scrlayer-scrpage` les paqs obsolètes seront utilisés avec `scrpage2`, `scrpage` ou avec `fancyhdr`, pourtant peu compatible avec `KOMA-Script` et dont l'usage est déconseillé avec `scrletter`. En outre, `scrheadings` de la page de style est activé par le chargement automatique interne `scrlayer-scrpage`. Cela signifie que les mises en page `headings` et `plain` des `alias` de classes utilisés sont sous le contrôle de `scrlayer-scrpage`.

```
\pagestyle{letter}
\pagestyle{plain.letter}
\setkomavar{nexthead}[désignation]{contenu}
\setkomavar{nextfoot}[désignation]{contenu}
```

Le style de page lettre, défini dans `scrletter` en utilisant `scrlayer-scrpage`, est indépendant des styles de page de la classe `letter`. Les variables sont `nexthead` et `nextfoot` pour la lettre ainsi utilisée, comme cela est documenté dans la section 4.13 pour les rubriques de style de page. Le style de page brut correspondant coïncide aussi avec le style de page `plain` de `scrletter2`. Ainsi, le numéro de page est placé selon l'option `pagenumber`.

```
headsepline
footsepline
```

Les deux options existantes dans `scrletter2`, `headsepline` et `footsepline` peuvent avec `scrletter` obéir aux règles de `scrlayer-scrpage`. Cela concerne la définition des lignes haut et bas de page des styles `plain` et `plain.letter` (voir section 5.5).

```
\pagemark
\letterpagemark
```

Comme on pouvait déjà définir la classe `KOMA-Script` `\pagemark`, l'utiliser et donc affecter directement une redéfinition par `scrletter` non seulement à la pagination des lettres, mais à toutes paginations, `scrletter` ne peut pas facilement charger le numéro de page via `\pagename` précédé du terme « page ». Au lieu de cela, il définit `\letterpagemark` en conséquence et les ensembles `\letterpagemark` dans `\begin{letter}` pour la durée de la lettre à la définition de `\letterpagemark`.

Pour redéfinir avec `\letterpagemark` à la fois le document normal et les lettres, vous devez ajouter `\let\letterpagemark\pagemark` en supplément à la redéfinition normale de `\pagemark`. Cette déclaration n'interfère pas avec

le reste même, si vous transposez plus tard la lettre dans la classe `scr1ttr2`, `\letterpagemark` n'est ni connu, ni utilisé.

## 21.4 Le traitement des fichiers avec `lco-scrletter`

Il est dit, section 21.4 que vous pouvez charger directement des fichiers `.ico` via `\documentclass`. Avec `scrletter`, il faut renoncer à cette possibilité.

```
\LoadLetterOption{nom}  
\LoadLetterOptions{liste des noms}
```

S'il est recommandé seulement pour `scr1ttr2` de charger des fichiers `ico` avec `\LoadLetterOption` ou `\LoadLetterOptions`, c'est obligatoire pour `scrletter`. Bien entendu, les fichiers `ico` peuvent également être chargés après `scrletter`.

## 21.5 Fichiers `.ico` pour utilisateur avancé

Bien que tous les formats de `typearea` soient utilisés, il est possible que, à la sortie de la première page d'une lettre, certains formats conduisent à des résultats indésirables. Ce n'est pas le concept de classe, mais des paramètres pour ISO A4 existent et il n'y a pas, malheureusement, de règles générales qui permettent le calcul des positions de champ d'adresse et autres dans chacun des formats de papier. Cependant, il est possible de définir des ensembles de paramètres supplémentaires pour d'autres s de papier

### 21.5.1 Suivi de la taille du papier

Actuellement, il existe des jeux de paramètres et fichiers `.ico` pour format A4 et format lettre. Toutefois, la classe `scr1ttr2` comprend beaucoup plus de formats de papier dans la théorie. Par conséquent, il est nécessaire de contrôler si un format de papier correct est réglé :

```
\LetterOptionNeedsPapersize{nom d'option}{taille du papier}
```

Vous serez averti lorsque, dans le fichier `.ico`, un format de papier non prévu est livré avec KOMA-Script et `\LetterOptionNeedsPapersize` a besoin des instructions fournies par le fichier `.ico` pour trouver l'option `format`.

Le premier argument est le nom du fichier `.ico` passé sans le suffixe `.lco`.

Le second argument, la taille du papier est passé par ce fichier lco conçu pour.

Plusieurs fichiers .ico successifs sont chargés et chacun de ces fichiers contient une commande `\LetterOptionNeedsPapersize`. La commande `\opening` ne fera que tester la dernière taille de papier spécifié. Il est par conséquent possible, pour l'utilisateur expérimenté, d'écrire facilement des fichiers .ico avec des ensembles de paramètres pour d'autres formats de papier.

Vous utilisez un papier A5 avec orientation normale pour vos lettres ? vous le mettez dans une enveloppe format C6 à fenêtre standard ? la position du champ d'adresse sera en principe la même qu'avec une lettre normale, conformément à DIN A4. La différence est que, en substance, le papier A5 ne nécessite qu'un seul pliage, évitant les marques de pliage supérieure et inférieure. Ceci peut être réalisé, par exemple, en plaçant les marques à l'extérieur du papier :

```
\ProvidesFile{a5.lco}
      [2015/03/21 letter class option]
\LetterOptionNeedsPapersize{paper=a5}{a5}
\@setlength{tfoldmarkvpos}{\paperheight}
\@setlength{bfoldmarkvpos}{\paperheight}
\endinput
```

Il serait élégant de désactiver les marques utilisant les options `foldmarks`. En outre, les pseudo-longueurs `firstfootvpos` sont réglables en cas de besoin pour la mise en place du pied de page. Je laisse au lecteur le soin de trouver une valeur appropriée. Avec un tel fichier .ico, il est seulement important que d'autres options de fichier .ico, comme `SN` soient précisées avant le chargement de `a5.ico`.

## 21.5.2 Visualisation de positions

Si vous développez vous-même des fichiers .ico, par exemple les différents champs de l'entête pour les adapter à vos besoins, il sera sans doute utile de visualiser au moins certains éléments directement. À cette fin, il existe le fichier `visualize.lco` qui peut être chargé comme n'importe quel fichier .ico, mais l'impact de cette option `scrletter` est limité au préambule du document et ses effets ne seront pas annulés. Le fichier .ico utilise les paqs `eso-pic` et `graphicx`, qui n'appartiennent pas à KOMA-Script.

```
\showfields{liste de champs}
```

## 21 Informations supplémentaires

---

Cette déclaration permet d'activer la visualisation des champs de l'entête. La liste des arguments de champ est une liste des champs à visualiser séparés par des virgules. Les champs de la table 21.2 sont supportés.

TABLE 21.2 – Liste d'arguments de champs

<code>test</code>	boîte de test mesurant 10 par 15 cm, et qui est à 1 cm du bord supérieur gauche du papier. Ce champ de test existe à des fins de débogage. Il sert comme mesure de comparaison dans le cas où dans le processus de création de documents, les mesures sont corrompues.
<code>head</code>	domaine de l'entête de la feuille de papier à lettre, champ ouvert vers le bas.
<code>foot</code>	domaine du pied de page de la feuille de papier à lettre, champ ouvert vers le bas.
<code>address</code>	fenêtre d'adresse
<code>location</code>	domaine d'extension de l'expéditeur.
<code>refline</code>	occupation de ligne, champ ouvert vers le bas.

En utilisant les commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9) pour l'élément de champ, la couleur de visualisation est changée. La valeur par défaut est `\normalcolor`.

```
\setshowstyle{style de visualisation}
\edgesize
```

Par défaut, les champs individuels sont marqués par des lignes de limites. Cela correspond à la trame de style de visualisation. Les champs ouverts vers le bas ne sont pas entièrement encadrés, mais affichés avec de petites flèches orientées vers le bas. La règle de style de visualisation est également disponible comme alternative, auquel cas, le champ est mis en évidence sans que l'on puisse différencier s'il est ouvert ou fermé vers le bas. Les champs ouverts sont affichés avec une hauteur minimale vers le bas. Le troisième style de visualisation disponible est la marge. Les coins des champs sont mis en évidence. Dans les champs ouverts vous pouvez passer des marqueurs de coin inférieurs. La taille des marqueurs de coin est stockée dans la macro `\edgesize` avec un pré-réglage de 1 ex.



```

\showenvelope(largeur, hauteur)(HOffset, VOffset) [supplément]
\showISOenvelope{format}[supplément]
\showUScommercial{format}[supplément]
\showUScheck[supplément]
\unitfactor

```

Ces instructions sont utilisées pour présenter en sortie une page comportant le dessin d'une enveloppe toujours éditée avec une rotation de 90 ° sur sa propre page à l'échelle 1 : 1.

La fenêtre positionne automatiquement l'adresse sur la feuille de papier à lettre à partir des données actuelles générées par `toaddrvpos`, `toaddrheight`, `toaddrwidth` et `toaddrhpos`. À cet effet, il est nécessaire de connaître la valeur pour laquelle la feuille de lettre pliée est plus petite de chaque côté que la largeur et la hauteur de l'entête. Si ces deux valeurs, `HOffset` et `VOffset`, ne sont pas spécifiées avec `\showenvelope`, une tentative sera calculée à partir des repères de pliage et de la taille du papier lui-même. Les instructions `\showISOenvelope`, `\showUScommercial` et `\showUScheck` sont basées sur `\showenvelope`.

Avec `\showISOenvelope`, une enveloppe ISO C4, C5, C5/6, DL (également connu sous le nom C5/6) ou C6 est générée. Avec `\showUScommercial`, c'est une enveloppe commerciale américaine qui sort au format 9 ou 10. `\showUScheck` est responsable en dernier ressort des enveloppes dans le format de chèque US. Dans l'enveloppe, la position de la feuille de papier à lettre est indiquée par des lignes pointillées.

La couleur utilisée dans ce cas peut, à l'aide des commandes `\setkomafont` et `\addtokomafont` (voir rubrique 4.9), être modifiée pour cet élément de la lettre, sa valeur par défaut est `\normalcolor`. La couverture aura son dessin calibré automatiquement. La dimension se fait par multiples de `\unitlength` avec un maximum de précision de  $1/\text{\unitfactor}$ , l'exactitude du calcul de  $\text{\TeX}$  et la taille de l'étiquette en sont les limites réelles. Sa valeur de 1 par défaut peut être redéfinie avec `\renewcommand`.

Exemple : un modèle de lettre est créé au format ISO A4. Les champs doivent être soutenus afin de vérifier leur position marquée avec des bordures jaunes. En outre, vous voulez vérifier la position de la fenêtre dans une enveloppe de taille DL à l'aide d'un dessin. Les chiffres des lignes de dimensions sont en rouge, les valeurs numériques émises dans une police plus petite, et les mesures en cm avec une précision de 1mm. L'enveloppe de papier à lettre en pointillés devrait être de couleur verte :

```
\documentclass[visualize]{srlttr2}
```

```
\usepackage{xcolor}
\setkomafont{field}{\color{yellow}}
\setkomafont{measure}{\color{red}\small}
\setkomafont{letter}{\color{green}}
\showfields{head,address,location,refline,foot}
\usepackage[frenchb]{babel}
\usepackage{lipsum}
\begin{document}
\setkomavar{fromname}{Charles Legrand}
\setkomavar{fromaddress}{la foraisserie
                        41200 Pijonbeau l'étroit château}
\begin{letter}{\%
  Papiray
  chemin des vignes
  41250 Monvillage\%
}
\opening{bonjour,}
\lipsum[1]
\closing{à plus}
\end{letter}
\setlength
{\unitlength}{1cm}
\renewcommand*{\unitfactor}{10}
\showISOenvelope{DL}
\end{document}
```

Cela montrera l'entête sur la première page et le graphisme de l'enveloppe sur la deuxième page du document. En ce qui concerne la dimension il est important de noter que les combinaisons défavorables de `\unitlength` et `\unitfactor` peuvent provoquer une erreur  $\TeX$  de type dépassement arithmétique. Il peut aussi arriver que les paramètres de sortie diffèrent légèrement de la valeur réelle. Aucune de ces erreurs n'est une erreur de visualisation mais signale seulement des limites spécifiques à l'implémentation.

## 21.6 Prise en charge linguistique

La classe `scrlltr2` supporte de nombreuses langues, dont *german* pour la vieille orthographe allemande et *ngerman* pour la nouvelle, *austrian* intégrant

l'autrichien avec l'ancienne orthographe allemande et la nouvelle *naustrian*, et aussi *english* sans indication pour américain et britannique ou américain, puis *USenglish* pour américain ou britannique et *UKenglish* pour britannique, et français, italien, espagnol, néerlandais, croate, finnois, norvégien, suédois, polonais, tchèque, slovaque...

Basculer entre les langues en utilisant l'extension `babel` (voir [Bra01]) avec la commande `\selectlanguage{langue}`.

D'autres paqs tels que l'allemand (voir [Rai98a]) et *ngerman* (voir [Rai98b]) appliquent également cette directive. Mais en général, la sélection de la langue a déjà été faite lors de l'initialisation du paq.

Permettez-moi une note sur les paqs de langage à commuter. Le paq français (voir [Gau07]) gère non seulement la redéfinition des critères dans la table 21.1 page 574 mais aussi d'autres changements. C'est la définition à propos de la déclaration `\opening`. Il est simplement supposé que `\opening` est toujours défini dans la classe lettre `standard`, ce qui n'est pas le cas avec `scr1ttr2`.

Mais le paq français détruit la définition de `scr1ttr2` et ne fonctionne pas correctement avec KOMA-Script.

Je vois cela comme un échec du paq français. Si le paq de `babel` utilisé pour passer à la langue française est également le paq français, il est probable que le résultat entraîne exactement les mêmes problèmes.

Avec la version 3.7j de `babel`, ce problème ne se pose, cependant, que s'il est spécifié explicitement par l'option que `babel` doit utiliser le paq français.

Comme on ne peut être sûr de la version de `babel`, je recommande d'utiliser `\usepackage [frenchb]{babel}` pour sélectionner la langue française. Le cas échéant, vous pourrez ensuite passer avec `\selectlanguage{french}` pour sélectionner le français.

Il est possible que des problèmes similaires existent avec d'autres langues et paqs. Si la langue d'une lettre change, les critères de la table 21.5 page 599 sont redéfinis sur ces instructions. Si la langue utilisée pour la commutation de paq ne le supporte pas, les déclarations ci-dessus peuvent être utilisées directement en cas de besoin.

Les dates des données numériques (voir option `numericaldate` à la section 4.10) prennent une forme différente selon le langage utilisé. Les informations en détail peuvent être trouvées dans la table 21.3 page suivante. Les instructions doivent contenir le terme spécifique à la langue respective. Celui-ci peut être

## 21 Informations supplémentaires

---

personnalisé pour soutenir une nouvelle langue ou même pour vos loisirs. Le procédé de mise en place est expliqué à la [section 11.4](#).

TABLE 21.3 – Formats de sortie pour la date spécifiques à la langue

Langue	Commande	Date exemple
<code>\captionssacadian</code>	<code>\dateacadian</code>	24/12/1993
<code>\captionssamerican</code>	<code>\dateamerican</code>	12/24/1993
<code>\captionssaustralien</code>	<code>\dateaustralien</code>	24/12/1993
<code>\captionssaustrrian</code>	<code>\dateaustrrian</code>	24.12.1993
<code>\captionssbritish</code>	<code>\datebritish</code>	24/12/1993
<code>\captionsscanadian</code>	<code>\datecanadian</code>	1993/12/24
<code>\captionsscanadien</code>	<code>\datecanadien</code>	1993/12/24
<code>\captionsscroatian</code>	<code>\datecroatian</code>	24.12.1993.
<code>\captionssczech</code>	<code>\dateczech</code>	24.12.1993
<code>\captionssdutch</code>	<code>\datedutch</code>	24.12.1993
<code>\captionssenglish</code>	<code>\dateenglish</code>	24/12/1993
<code>\captionssfinnish</code>	<code>\datefinnish</code>	24.12.1993.
<code>\captionssfrançais</code>	<code>\datefrançais</code>	24.12.1993
<code>\captionssfrench</code>	<code>\datefrench</code>	24.12.1993
<code>\captionssgerman</code>	<code>\dategerman</code>	24.12.1993
<code>\captionssitalian</code>	<code>\dateitalian</code>	24.12.1993
<code>\captionssnaustrian</code>	<code>\datenaustrian</code>	24.12.1993
<code>\captionssnewzealand</code>	<code>\datenewzealand</code>	24/12/1993
<code>\captionssngerman</code>	<code>\datengerman</code>	24.12.1993
<code>\captionssnorsk</code>	<code>\datenorsk</code>	24.12.1993
<code>\captionssnswissgerman</code>	<code>\datenswissgerman</code>	24.12.1993
<code>\captionsspolish</code>	<code>\datepolish</code>	24.12.1993
<code>\captionsslovak</code>	<code>\dateslovak</code>	24.12.1993
<code>\captionssspanish</code>	<code>\datespanish</code>	24.12.1993
<code>\captionssswedish</code>	<code>\dateswedish</code>	24/12 1993
<code>\captionssswissgerman</code>	<code>\dateswissgerman</code>	24.12.1993
<code>\captionssUKenglish</code>	<code>\dateUKenglish</code>	24/12/1993
<code>\captionssUSenglish</code>	<code>\dateUSenglish</code>	12/24/1993

## 21.7 De `scrlettr` obsolète à l'actuel `scrlettr2`

TABLE 21.5 – Paramètres par défaut pour les termes dépendant des langues *english* et *ngerman*, tant que les paqs de sélection de langue n'ont pas été utilisés

Instructions	english	ngerman	français**
<code>\bankname</code>	<i>Bank account</i>	<i>Bankverbindung</i>	compte bancaire
<code>\ccname*</code>	<i>cc</i>	<i>Kopien an</i>	copie à
<code>\customername</code>	<i>Customer no.</i>	<i>Kundennummer</i>	numéro de client
<code>\datename</code>	<i>Date</i>	<i>Datum</i>	date
<code>\emailname</code>	<i>Email</i>	<i>E-Maile</i>	mail
<code>\enclname*</code>	<i>encl</i>	<i>Anlagen</i>	aménagements
<code>\faxname</code>	<i>Fax</i>	<i>Fax</i>	fax
<code>\headfromname</code>	<i>From</i>	<i>Von</i>	de
<code>\headtoname*</code>	<i>To</i>	<i>An</i>	à
<code>\invoicename</code>	<i>Invoice no.</i>	<i>Rechnungsnummer</i>	numéro de facture
<code>\myrefname</code>	<i>Our ref.</i>	<i>Unser Zeichen</i>	notre connexion
<code>\pagename*</code>	<i>Page</i>	<i>Seite</i>	page
<code>\phonename</code>	<i>Phone</i>	<i>Telefon</i>	téléphone
<code>\subjectname</code>	<i>Subject</i>	<i>Betrifft</i>	sujet
<code>\wwwname</code>	<i>Url</i>	<i>URL</i>	url
<code>\yourmailname</code>	<i>Your letter of</i>	<i>Ihr Schreiben vom</i>	Votre lettre du
<code>\yourrefname</code>	<i>Your ref.</i>	<i>Ihr Zeichen</i>	Votre référence

Le terme dans `scrlettr2` doit être réglé seulement après le préambule, c'est-à-dire après `\begin{document}`. Il n'est donc pas disponible auparavant et ne peut être modifié avant.

Les valeurs par défaut se trouvent dans la table 21.5.

\* Normalement, ces termes sont définis par les paqs de langues telles que `babel`. Dans ce cas, ils ne sont pas redéfinis par `scrlettr2` et peuvent différer du tableau ci-dessus.

\*\* La colonne en français est destinée à une meilleure compréhension du texte précédent.

## 21.7 De `scrlettr` obsolète à l'actuel `scrlettr2`

L'ancienne classe de lettre devenue obsolète en 2002 avec l'introduction de `scrlettr2` (voir chapitre 4) n'est plus soutenue et ne devrait plus être utilisée

mais sa documentation est accessible dans `scrlettr.dtx`. Les meilleurs résultats avec  $\LaTeX$  sont obtenus grâce à ses fichiers, citons par exemple :

- $\LaTeX$  `scrlettr.dtx`,
- `mkindex scrlettr`,
- $\LaTeX$  `scrlettr.dtx`,
- `mkindex scrlettr`,
- $\LaTeX$  `scrlettr.dtx`,

Le fichier `scrlettr.dvi` est obtenu avec ces instructions et  $\LaTeX$  remplace `\pdf1.pdf`. Afin de faciliter la transition de l'ancienne à la nouvelle classe, il existe avec `KOMAold` un cadre de compatibilité. Fondamentalement, l'ensemble des fonctionnalités d'origine est inclus dans la nouvelle classe.

```
\yourrefname
\yourmailname
\myrefname
\customername
\invoicename
\subjectname
\ccname
\enclname
\headtoname
\headfromname
\datename
\pagename
\mobilephonenumber
\phonenumber
\faxname
\emailname
\wwwname
\bankname
```

Toutefois, sans `KOMAold`, l'interface utilisateur et les paramètres par défaut ne correspondent pas l'un à l'autre. Pour plus de détails, voir la table 21.5 page précédente, section 4.21, ainsi que le livre `KOMA-Script` [KM12].

# Chapitre 22

## Création de répertoires avec `splitidx` et `SplitIndex`

### 22.1 Résumé

Il existe un paq standard de L<sup>A</sup>T<sub>E</sub>X, `makeidx`, pour créer un index-répertoire<sup>1</sup> dans chaque classe de document. Mais il est parfois nécessaire de créer plus d'un index et, pour ce faire, il existe des paqs différents avec des solutions différentes aux problèmes différemment posés pour en générer plusieurs. Le paq `splitidx`<sup>2</sup> propose la mise en œuvre d'une nouvelle solution de prise en charge de ces problèmes. En outre, `splitidx` permet de personnaliser la mise en page, l'apparence et la mise en forme des entrées individuelles de répertoires.

---

1. PAPIRAY précise : dans ce chapitre, j'utilise répertoire et catalogue pour désigner l'index indifféremment.

2. Ce chapitre est la version v1.2c de `splitidx.dtx`. Ce paq est également une création de Markus KOHM mais il ne figure pas, à ce jour, dans `KOMA-Script`. Bien que mis en circulation en date du 18/02/2016, cette version devrait néanmoins être stable. Merci à Michael PALMER pour avoir développé le mode d'emploi en anglais et à Raymond ROCHEDIEU à qui l'on doit la version française de ce document.

## 22.2 Introduction

La norme  $\text{\LaTeX}$  ne prévoit qu'un seul répertoire de document et, pour le produire, on charge `makeidx` qui marque les entrées d'index dans le document à l'aide de la commande `\index`. Lorsque le document est traité avec  $\text{\LaTeX}$ , les commandes `\index` sont écrites en tant que commandes `\indexentry` à l'état brut dans le fichier répertoire `\jobname.idx`<sup>3</sup>. Le fichier répertoire brut est ensuite traité avec `MakeIndex` ou un autre programme auxiliaire tel que `xindy`, qui produira un fichier répertoire nommé `\jobname.ind`. Ce fichier est ensuite inclus à la fin du document en utilisant la commande `\printindex`.

Le paq `splitidx` étend ce processus à la création de répertoires multiples. Des répertoires distincts sont déclarés avec un identifiant raccourci unique donné par la commande `\newindex`. Dans le document, les entrées individuelles sont marquées et affectées à des répertoires spécifiques avec la commande `\sindex`. Pour chaque répertoire déclaré, un fichier séparé `.idx` est généré, et chacun est en post-traitement dans un fichier séparé `.ind` qui est ensuite inclus dans le document à l'aide d'une version modifiée de la commande `\printindex`<sup>4</sup>. Le processus décrit ressemble à celui d'autres paqs multi-répertoires tel que `multind`. La façon la plus simple de mettre en œuvre ce régime est celle utilisée par le paq `multind`, c'est-à-dire écrire directement un fichier `.idx` séparé pour chaque répertoire déclaré lors du traitement du document avec  $\text{\LaTeX}$ , puis post-processus, traiter chaque fichier `.idx` à part avec `MakeIndex`. Cependant, cette approche peut fonctionner dans des limites techniques.

$\text{\TeX}$  ne peut pas avoir plus de 16 fichiers ouverts en écriture à un moment donné. Plusieurs de ces fichiers sont tenus, à d'autres fins telles que références croisées ou table des matières en fonction de la structure de votre document, par  $\text{\LaTeX}$  lui-même. Par conséquent, si vous avez besoin d'un grand nombre de répertoires distincts, le nombre limité de fichiers traités peut devenir un problème auquel le paq `splitidx` apporte une solution.

Si ce nombre peut être incorporé à l'intérieur du nombre de répertoires traités, vous pouvez utiliser `splitidx` avec l'option de paq partagé. Ensuite, `splitidx` va écrire directement l'index brut de fichiers multiples qui se comportera selon

---

3. « `jobname` » est le nom générique, en principe le nom du fichier principal, créé lors de la compilation et utilisé dans les fichiers « `.idx` » et « `.ind` »

4. Pour plus de détails sur le post-traitement du fichier intermédiaire, lire [1] de Leslie LAMPORT : *MakeIndex - An Index Processor For  $\text{\LaTeX}$* , 17 February 1987 et par exemple [2] Pehong CHEN, Rick P. C. RODGERS : *MAKEINDEX(1L)*, Manual page, 10 December 1991.



le régime décrit. D'autre part, si le nombre de répertoires dépasse le nombre de fichiers gérés, vous pouvez demander à `splitidx` d'écrire les entrées d'index avec un seul fichier répertoire intermédiaire qui doit ensuite être post-traité afin d'obtenir des fichiers séparés faits avec le programme `SplitIndex` qui existe dans plusieurs implémentations différentes (voir ci-dessous). Ce comportement de `splitidx` est activé en omettant l'option `split`, ce qui est le comportement par défaut du `paq`. En plus de la construction de répertoires distincts, `splitidx` offre une aide à la personnalisation de la composition et de l'apparence de ces répertoires, ainsi que la mise en forme des entrées individuelles.

## 22.3 Le programme *SplitIndex*

### 22.3.1 Objectif

Alors que le nombre de fichiers que `TEX` peut ouvrir pour l'écriture est limité, l'utilisation de plusieurs index est normalement limitée aussi. Comme il est déjà mentionné dans l'article 1, cette limitation peut être neutralisée à l'aide d'un seul fichier répertoire intermédiaire qui sera divisé en plusieurs fichiers index bruts par un post-processeur externe : `SplitIndex`.

### 22.3.2 Mise en œuvre

Le programme a été mis en œuvre dans cinq langages différents, comme suit :

`splitindex.pl` : version écrite en perl, vous avez besoin d'un interpréteur perl pour l'exécuter. Si vous êtes un utilisateur Unix, vous avez cet interpréteur et vous pouvez appeler `splitindex.pl` comme vous appelleriez un programme binaire ou un script shell. C'est la mise en œuvre de référence, celle que je préfère utiliser, parce que qu'elle fut la première, la plus facile et la plus courte à écrire.

`splitindex.java` : cette version que j'ai écrite en utilisant Sun Java 1.4.1 car Java peut être installé partout et que beaucoup de gens en comprennent les fichiers source. Néanmoins, il n'y a plus de pré-version de la compilation dans la distribution principale que vous pouvez télécharger à partir du dépôt sur :

```
http://sarovar.org/plugins/scm cvs/cvsweb.php/binaries/?cvsroot=splitindex
```

`splitindex.c` : cette version est un source C de `splitindex` que j'ai écrit parce que la plupart des auteurs de logiciels comprennent le langage « C », et que certaines personnes veulent un binaire plus rapide que le code Java, même si ce dernier reste assez rapide. Cependant, il n'y a pas de binaires plus longs produits à partir de cette source dans la distribution principale.

`splitindex.tex` : version T<sub>E</sub>X du programme. Oui, vous avez raison : il existe une plate-forme dans T<sub>E</sub>X mais qui n'a pas l'ensemble des fonctionnalités de ce programme (voir paragraphe 3.13), indépendant du système et qui ne nécessite pas d'installer `perl` ou Sun Java 1.4. Il n'est pas impossible d'en fixer tous les inconvénients, mais il est plus facile d'utilisation et accomplit beaucoup plus de travail que tous ses homologues.

`splitindex.tlu` : une nouvelle version du programme de LuaT<sub>E</sub>X, plate-forme indépendante comme le script `perl`. Noter que la syntaxe des expressions régulières diffère de la syntaxe de `perl`, si vous l'utilisez au lieu de cette version. Les distributeurs devraient préférer la version `perl` si elle est également fournie pour la plate-forme d'installation.

À l'exception de la version de T<sub>E</sub>X, tous ces programmes sont également en mesure d'appeler le processeur de répertoire sur chacun des fichiers d'index brut obtenus.

Et où sont le zézaiement, le papotage, le prologue, la version de `splitindex`? Hey, cinq langages sont suffisants pour moi! Si vous avez besoin d'un langage supplémentaire, écrivez-le!<sup>5</sup>

## 22.4 Utilisation du paq `splitidx`

### 22.4.1 Installation

Vous pouvez utiliser `splitidx` comme substitut pour `makeidx`. Si vous le faites, vous devez remplacer

```
\usepackage{makeidx}
par
\usepackage{splitidx}
```

---

5. PAPIRAY précise : cette remarque et cette invitation, auxquelles je n'ai ni ajouté, ni retranché une virgule, sont dues en totalité à l'auteur Markus KOHM.

Pour activer la génération de répertoire, vous pouvez utiliser `\makeindex`, qui est déclaré par le noyau L<sup>A</sup>T<sub>E</sub>X. Vous pouvez également charger le paq avec l'option `makeindex` :

```
\usepackage[makeindex]{splitidx}
```

qui est presque le même que :

```
\usepackage{splitidx}\makeindex
```

D'autres paqs-options sont disponibles. L'effet de l'option `split` a déjà été décrit dans la [section 1](#) ; d'autres options sont présentées ci-après.

Pour générer plus d'un répertoire sans raccourci, vous devriez le déclarer en utilisant `\newindex` avec la syntaxe :

```
\newindex[nom de l'index]{raccourci}
```

L'argument obligatoire `shortcut` est utilisé pour citer les index, voir à ce sujet la description de `\sindex` pour plus d'informations. L'argument `index name` facultatif est le nom du répertoire. Il est utilisé par défaut par les macros `\printindex` et `\printsubindex` (voir ci-dessous). Si vous oubliez ce nom, le raccourci sera utilisé comme nom du répertoire. Alors qu'il est toujours bon de déclarer tous les répertoires dans le préambule du document, cela doit être également fait si vous utilisez le paq fractionné. Dans ce cas, la commande `\newindex` ouvre un fichier index brut à écrire pour chaque répertoire déclaré. Comme seule exception, le fichier brut pour les entrées avec le raccourci `idx` sera créé par défaut automatiquement et le nombre de répertoires est limité par le nombre limité de flux d'édition fournis par T<sub>E</sub>X. Si vous dépassez ce nombre, non seulement la macro `\newindex` peut entraîner une erreur, mais aussi `\tableofcontents`, `\listoffigures`, `\listoftables` et toute autre commande qui attribue un flux d'édition.

Un raccourci unique, déclaré avec `\newindex` se réfère à un index spécifique et devient une partie du nom de fichiers correspondants à `et` et `.ind`. Par conséquent, lorsque vous choisissez un raccourci, assurez-vous que vous utilisez uniquement des caractères ou des symboles qui sont autorisés dans les noms de fichiers. Dans le système qui traite les noms de fichiers comme insensibles à la casse, vous ne devriez pas mélanger les lettres majuscules et minuscules. Pour une portabilité maximum et un minimum de problèmes, il est préférable d'utiliser uniquement des lettres minuscules.

## 22.4.2 Marquage des entrées d'index

Après le chargement du paq `splitidx`, utiliser la commande `\index` pour marquer les entrées de répertoires dans votre manuscrit comme d'habitude, description de l'argument et les caractéristiques de cette commande dans la référence [1]<sup>6</sup>. Le programme de `splitindex` (voir le paragraphe 3.10) va mettre toutes les entrées de répertoires produites avec `\index` dans le même fichier index brut étiqueté avec le raccourci unique `.idx`; cependant, cette commande `\index` ne permet pas d'attribuer des entrées séparées. L'option `useindex`, décrite ci-après, permet de modifier ce comportement.

Le paq `splitidx` définit la commande `\sindex` avec la syntaxe :

```
\sindex[raccourci]{répertoire-entrée}
```

La commande `\sindex` est le mécanisme `splitidx` pour placer les entrées d'index individuels dans des répertoires spécifiques, identifiés en passant son raccourci unique comme il a été déclaré, avec `\newindex`, dans l'argument de `\sindex`. Si rien n'est spécifié, ce sont les valeurs par défaut de raccourci `.idx` qui devraient être utilisées pour identifier une sorte d'index général.

Si vous le souhaitez, vous pouvez également demander que `\index` soit un alias pour `\sindex`. Pour ce faire, vous utilisez l'option du paq `useindex` par exemple :

```
\usepackage[useindex]{splitidx}
```

Cela peut être utile lors de l'utilisation de paq comme `jurabib` qui attend que `\index` soit une commande de répertoire.

## 22.4.3 Suppression de la génération de répertoires multiples

Dans certaines circonstances, par exemple lorsque l'on travaille avec un éditeur qui applique un format de document rigide, il peut être nécessaire de fusionner les répertoires distincts en un seul. Dans ce cas, retirer toutes les marques individuelles jusqu'à l'index identifiant, ou utiliser l'alternative consistant à charger le paq `splitidx` avec l'option `allintoone` :

```
\usepackage[allintoone]{splitidx}
```

---

6. Leslie LAMPORT : *MakeIndex - An Index Processor For L<sup>A</sup>T<sub>E</sub>X*, 17 February 1987, ouvrage déjà cité.

ou

```
\usepackage[allintoone,makeindex]{splitidx}
```

Avec cette option, *splitidx* épluche le programme pour vous et remplace `\sindex[raccourci] {index d'entrée}` par `\index{indexentry}` pendant l'action de L<sup>A</sup>T<sub>E</sub>X.

Note : Actuellement, seule l'une des options *allintoone* ou *useindex* peut être utilisée au même moment. Si vous essayez d'utiliser les deux, *useindex* sera désactivé. Il peut en résulter, dans ce cas, de nombreux messages d'erreur.

#### 22.4.4 Personnalisation des entrées de répertoires

*splitidx* utilise `\protected@write` pour écrire les entrées de répertoires à l'édition de fichiers. La macro `\AtWriteToIndex` permet d'exécuter un morceau de code à chaque fois qu'un index est écrit pour un répertoire spécifique :

```
\AtWriteToIndex{raccourci}{code}
```

Cela peut être utile si vous voulez que vos entrées de répertoires ne référencent pas le numéro de page mais un autre compteur à la place. Par exemple, afin que chaque entrée du répertoire général (identifié par le raccourci *idx*) pointe vers le numéro de *section* correspondant, vous pouvez écrire

```
\AtWriteToIndex{idx}{\let\thepage\thesection}
```

Noter que cela ne fonctionnera que si le raccourci de l'index est donné dans chaque annotation d'entrée de répertoire ; par exemple,

```
\sindex[idx]{Planche à roulettes}
```

au lieu de

```
\sindex{Planches à roulettes}
```

Noter que si vous souhaitez utiliser la commande `\index` au lieu de `\sindex`, vous devriez aussi utiliser le paq-option *useindex*, sans quoi la commande `\index` va encore écrire le numéro de page du répertoire.

La commande `\AtWriteToIndex` ne peut être utilisée que dans le préambule du document.

Parfois, il peut être utile d'exécuter des commandes pour écrire une seule entrée. Pour ce faire, vous pouvez utiliser

```
\AtNextWriteToIndex{raccourci}{commande}
```

à la place de `\AtWriteToIndex`.

### 22.4.5 Commandes automatiques de répertoire personnalisé

Certaines personnes n'aiment pas écrire `\sindex[foo]{entrée}` mais veulent écrire `\foo{entrée}`. Pour ces personnes, l'option « `idxcommands` » a été mise en œuvre. Elle définit, pour chaque répertoire déclaré, une commande avec le nom du raccourci. Si vous utilisez cette option alors qu'une commande est déjà créée avec ce nom, vous obtiendrez une erreur. Noter également que si vous utilisez cette option, le raccourci ne doit être composé que de lettres.

### 22.4.6 Prévenir l'expansion prématurée des entrées

Lorsque vous utilisez le paq de répertoire `standard makeidx`, la commande du noyau L<sup>A</sup>T<sub>E</sub>X `\Index` peut étendre son argument. Le noyau utilise `\@sanitize` pour éviter l'expansion dans certains cas. Mais cela échoue si l'argument a déjà été lu par une autre macro.

Donc, si vous définissez une macro qui lit l'argument, l'utilise puis l'écrit dans le répertoire, cela peut le développer. Pour illustration, essayer ce qui suit :

```
\documentclass{article}
\usepackage{français}
\usepackage{makeidx}\makeindex
\newcommand*{\Test}[1]{#1\index{#1}}
\begin{document}
\Test{"Changement}
"Changement\index{"Changement}
\end{document}
```

Cela se traduira par deux entrées dans le fichier `.idx` :

```
\indexentry{\active@dq \dq@prtct{C}hangement}{1}
\indexentry{"Changement}{1}
```

Seule la première est quelque chose d'élargi qui n'a pas été cherché. Le paq `splitindx` se comporte, par défaut, de la même façon. Mais si vous utilisez `\newprotectedindex` pour définir un nouveau répertoire, il utilise une astuce pour éviter une expansion. Tous devraient se comporter comme cela et utiliser simplement le paq-option `protected` :

```
\documentclass{article}
\usepackage{français}
```

```

\usepackage[protected,useindex,makeindex]{splitidx}
\newcommand*\Test[1]{#1\index{#1}}
\begin{document}
\Test{"Changement}
"Changement\index{"Changement}
\end{document}

```

se traduira par deux entrées au fichier puis post-processus, traiter chaque fichier.idx :

```

\indexentry{"Changement}{1}
\indexentry{"Changement}{1}

```

Si vous voulez en savoir plus sur l'astuce, voir dans la documentation du noyau L<sup>A</sup>T<sub>E</sub>X, source2e la commande `\@onelevel@sanitize`.

### 22.4.7 Inclure les répertoires générés dans votre document

La commande `\printindex` est utilisée pour éditer un répertoire où tous les répertoires qui sont déclarés utilisent `\newindex` et dont le comportement dépend de la syntaxe utilisée. Par exemple, comme ceci :

```
\printindex[raccourci] [nom de l'index]
```

le fichier répertoire avec le raccourci optionnel sera inclus et édité avec le nom facultatif utilisé comme titre. Si le nom est omis, le nom du répertoire déclaré par défaut avec `\newindex` sera utilisé à la place. Si ce nom a été omis lui aussi, le raccourci lui-même sera utilisé comme titre.

Si les deux arguments optionnels, raccourci et nom de l'index, sont omis, vous utilisez simplement :

```
\printindex
```

cette commande se comporte comme `\printindex` du paq `makeidx` et ne devrait pas être utilisée avec plusieurs index. Vous pouvez également imprimer tous les index qui ont été déclarés en utilisant `\newindex` immédiatement. Utiliser la syntaxe :

```
\printindex*
```

pour faire cela. Les répertoires seront édités dans l'ordre dans lequel ils ont été déclarés à l'aide de :

```
\newindex.
```

### 22.4.8 Paramètres type des répertoires générés

`\printindex` utilise, par défaut, l'édition et le traitement de la classe active de répertoire. Habituellement, ce sera l'environnement `theindex`, mais ce n'est pas une certitude. Noter cependant que `\printindex` attend le nom du répertoire contenu dans la macro `\indexname`; sinon, il ne parviendra pas à composer le nom de l'index<sup>7</sup>.

La commande `\printsindex` est analogue à `\printindex` mais elle effectue quelques redéfinitions avant d'imprimer le répertoire :

- rétrograde la position du niveau du répertoire de 1, qui est le format du titre de répertoire utilisé par `\section*` au lieu de `\chapter*` avec des classes qui définissent `\chapter` (par exemple `book` ou `scrreprt`), et en utilisant `\subsection*` au lieu de `\section*` avec des classes qui ne gèrent pas `\chapter` (telle la classe `article`);
- désactive `\`, `\twocolumn` et `\clearpage`, `\cleardoublepage` qui sont par ailleurs utilisés pour démarrer une nouvelle page dans chaque répertoire;
- modifie le mécanisme de marque que `\markright` utilise pour la mise en place des entêtes en cours d'exécution au lieu de `\markboth`.

Avec cette macro, vous pouvez imprimer plusieurs répertoires dans le chapitre d'une classe avec `\chapter`, ou dans une `section`, si vous utilisez une classe sans `\chapter`.

Cette commande est le complément indispensable à l'utilisation d'une classe KOMA-Script. Le paquet `splitidx` la redéfinit comme suit :

```
\setindexpreamble[raccourci]{préambule}
```

Cela permet de définir un préambule distinct pour chaque répertoire. Note : Le paquet `splitidx` n'imprime pas le préambule lui-même. Au lieu de cela, avant de composer un répertoire avec un raccourci donné en utilisant `\printindex` ou `\printsindex`, il attribue le préambule défini pour ce raccourci à l'intérieur d'une macro `\index@preamble` dont la valeur peut être consultée, au niveau de l'utilisateur, avec la macro `\useindexpreamble` (voir ci-après).

Si vous avez défini votre propre environnement répertoire ou si vous prolongez l'existence d'un environnement `theindex` existant en utilisant `\extendtheindex`

---

7. Ce serait un échec de la classe ou du paquet utilisé, pas du paquet `splitidx`. Je ne sais pas quelle classe mais la documentation de la bibliothèque du paquet `tcolorbox` utilise `\kvtcb@text@index` au lieu de `\`. Depuis la version 1.2c `splitidx` redéfinit également `\kvtcb@text@index` localement.



ou autre, vous pouvez utiliser `\useindexpreamble` pour récupérer le préambule précédemment défini pour le répertoire courant à l'aide de `\setindexpreamble` :

```
\useindexpreamble[additional commands]
```

Cette macro ne se limite pas aux classes KOMA-Script mais peut également être utilisée, par exemple avec les classes normalisées. Les commandes passées dans l'argument supplémentaire ne sont utilisées que si le préambule actuel du répertoire est défini et non vide. Les auteurs des classes `Wrapper` peuvent les utiliser, par exemple pour ajouter des espaces verticaux supplémentaires après le préambule du répertoire si, et seulement si, un préambule est édité.

La macro `\indexshortcut` est seulement définie dans le corps de `\printindex` et de `\printsubindex`. Elle élargit le raccourci du répertoire spécifié qui est en cours d'édition. Ce dernier peut être utile lors de la définition de votre propre environnement ou de l'extension de l'environnement `theindex`, en utilisant par exemple `\extendtheindex`.

La plupart des classes définissent l'environnement `theindex` à utiliser pour l'édition du répertoire. Utiliser `\extendtheindex` avec cette syntaxe :

```
\extendtheindex{before begin}{after begin}{before end}{after end}
```

Vous pouvez prolonger cet environnement. Les commandes passées avant « commence » sont insérées dans `\begin{theindex}` après le démarrage du groupe, mais avant le code existant défini pour ce bloc de code ; passées après, elles seront insérées après `\begin{theindex}`. Et de la même manière, passées avant « la fin », elles sont insérées avant `\end{theindex}`, et passées après « la fin », elles sont utilisées dans `\end{theindex}` après la fin du répertoire, mais juste avant de terminer le groupe.

## 22.4.9 Exemples

Voyons comment nous pouvons obtenir plus d'un répertoire. Le texte de l'exemple est stupide ? alors oubliez-le en pensant à l'utilisation de `splitidx`.

```
\documentclass{article} % nous utilisons la classe article
\usepackage{splitidx} %... et le paq splitidx
\makeindex % et nous voulons générer un ou plusieurs répertoires
nous définissons 4 répertoires :
```

```
\newindex[répertoire général]{idx} %... 1er répertoire
\newindex[répertoire animaux]{ani} %... 2ème répertoire
\newindex[répertoire fruits]{fru} %... 3ème répertoire
```

```

\newindex[répertoire végétaux]{veg} %... 4ème répertoire

\begin{document}
Pommes\sindex\index{index!sindex}[fru]{pomme} % entrée dans le
                                                répertoire fru
Oranges\sindex\index{index!sindex}[fru]{orange} % entrée dans le
                                                répertoire fru
sont des fruits\sindex\index{index!sindex}{fruits}. % entrée
                                                implicite dans idx
Les roses\sindex\index{index!sindex}[veg]{rose} % entrée dans le
                                                répertoire veg
sont des vegetaux\index{vegetaux}. % entrée implicite dans idx
Les chats\sindex\index{index!sindex}[ani]{chat} % entrée dans le
                                                répertoire ani
sont des animaux\sindex\index{index!sindex}[idx]{animaux}. %
                                                entrée dans idx

\printindex* % éditer tous les répertoires
\end{document}

```

Après avoir traité le fichier ci-dessus avec  $\text{\LaTeX}$ , vous obtiendrez un fichier brut de répertoires avec le contenu suivant :

```

\indexentry[fru]{pomme}{1}
\indexentry[fru]{orange}{1}
\indexentry{fruits}{1}
\indexentry[veg]{rose}{1}
\indexentry[idx]{vegetaux}{1}
\indexentry[ani]{chat}{1}
\indexentry[idx]{animaux}{1}

```

La **section 3.10** explique comment convertir ce fichier intermédiaire séparé en fichiers répertoires bruts et en fichiers répertoires. Dans l'exemple ci-dessus, les quatre fichiers répertoires sont entrés avec une seule commande `\printindex*`. Chaque fichier produira une seule **section** qui commence sur une nouvelle page. Les titres de **section**, « Index général », « Index animaux », « Index fruits » et « Index végétaux » seront édités en mode `onecolumn`, suivis par les entrées de répertoires édités en mode `twocolumn`. Vous souhaitez peut-être formater tous les répertoires comme paragraphes d'une **section**? Vous pouvez le faire en remplaçant la commande `\printindex*` dans l'exemple ci-dessus, comme suit :

```

\twocolumn[% définir le titre d'une colonne
\section*{préface} % la section avec les répertoires %
\markboth{préface}{préface} % édition du titre en cours %
] % seulement avec deux colonnes de répertoire
\printsubindex* % édite tous les répertoires

```

Noter que j'ai utilisé `\printsubindex*` au lieu de `\printindex*` dans cette modification de l'exemple.

Nous passons maintenant aux entêtes en cours d'exécution pour les pages de répertoire. Si vous utilisez la page de style `plain` qui est par défaut celle de la classe d'articles, les entêtes en cours d'exécution sont vides, de sorte que vous ne devez pas les mettre en place. Toutefois, en utilisant les titres de style de page `headings` et la commande `\section*` pour formater les rubriques de plusieurs répertoires, vous devez mettre en place le fonctionnement des entêtes pour correspondre au titre courant du répertoire. Avec une classe *KOMA-Script*, vous pouvez utiliser `\addsec` ou `\addsec*` au lieu de `\section*` pour formater les titres, auquel cas vous n'aurez pas besoin de mettre à jour manuellement les entêtes en cours d'exécution.

Vous voulez que le répertoire général soit la `section`, tandis que les autres en seraient les `sous-sections`? remplacer le code précédent par :

```

##### Ce ne sera pas la chose que vous vouliez!#####
\printindex[idx] % édite le répertoire article idx
\printsubindex[ani] % édite le répertoire paragraphe ani
\printsubindex[fru] % édite le répertoire paragraphe fru
\printsubindex[veg] % édite le répertoire paragraphe veg

```

Cela aboutira à une section `twocolumn` contenant l'index général (identifié par `idx`) et trois `sous-sections` de contenant les autres index et une page `break` après l'index général. Pourquoi est-ce ainsi? LaTeX passe en mode `twocolumn` alors qu'il entre dans l'environnement `theindex` (qui est créé par la commande `\printindex`) et revient au mode `normal` lorsqu'il sort de `theindex`. Si le mode `twocolumn` était actif avant `\printindex`, une commande `\clearpage` sera émise à la fin de l'index.

Alors quelle est la solution? En vous souvenant de la commande `\extendtheindex`, vous pouvez écrire :

```

\begingroup % conserver l'extension locale suivante à ce groupe
\extendtheindex% des changements de l'environnement theindex
{}% pas de changement avant de commencer
{}% pas de changement après avoir commencé

```

```

    {\let\onecolumn\relax % désactiver \onecolumn, puis terminer
\let\clearpage\relax % désactiver \clearpage, puis terminer
    }% changer avant de terminer
    {}% ne pas changer avant de terminer
\printindex[idx] % édite le répertoire idx de l'article
\endgroup % fin du groupe extension de theindex
\printsubindex[ani] % édite le répertoire ani du paragraphe
\printsubindex[fru] % édite le répertoire fru du paragraphe
\printsubindex[veg] % édite le répertoire veg du paragraphe
\onecolumn % termine les répertoires

```

Avec cette extension, l'ensemble du répertoire sera réglé en mode `twocolumn`, sans page `break` avant le premier paragraphe. Cependant, vous devez revenir en mode `manuellement` à la fin des répertoires. L'exemple ci-dessus peut être modifié comme suit pour obtenir un répertoire de type `onecolumn` :

```

\begingroup % conserver l'extension locale au groupe
\makeatletter % permet l'ajout de @ au nom de macro
\extendtheindex% changement de l'environnement theindex
    {\let\twocolumn\@firstoftone % désactiver \twocolumn
    \let\onecolumn\@firstoftone % désactiver \twocolumn
    \let\clearpage\relax % désactiver \clearpage
    }% changer avant de commencer
    {}% ne pas changer avant de commencer
    {}% ne pas changer avant de terminer
    {}% ne pas changer après avoir terminé
\makeatother % désactive \makeatletter
\printindex % éditer le répertoire
\endgroup % fin du groupe extension theindex

```

Cela fonctionne non seulement avec des répertoires `splitted` mais aussi avec un index unique. La section suivante montre comment générer des répertoires distincts à partir d'un seul fichier intermédiaire.

### 22.4.10 Fractionnement des fichiers répertoires

En général, il est suffisant d'utiliser un paramètre pour donner un nom à l'un des programmes de répertoire intermédiaire de `splitindex` en divisant le fichier intermédiaire en plusieurs fichiers bruts, puis appeler `MakeIndex` sur chacun de ceux-ci. Le programme `splitindex` peut être chargé afin d'utiliser un autre

traitement de répertoire tel que `xindy`, ou fournir des options supplémentaires avec le traitement, par exemple « `-g` » pour utiliser le tri avec `MakeIndex` en allemand. Bien que le besoin en soit rare, il est également possible de modifier l'analyse du fichier de ce répertoire intermédiaire, la génération des noms de fichiers et le contenu brut du répertoire des fichiers résultants.

Le nom des options et la syntaxe des arguments sont les mêmes dans tous les programmes à l'exception de `splitindex.tex` (voir le paragraphe 3.13) :

- `--help` (assistance)
  - `-h` affiche les informations sur l'utilisation, les options et arguments et termine sans traitement un fichier répertoire.
- `--makeindex nom de programme`
  - `-m nom de programme` appelé au lieu de `makeindex` pour traiter chaque fichier répertoire brut généré. Vous pouvez définir cette variable avec une valeur vide. La façon de le faire dépend du shell que vous utilisez, par exemple avec `bash` la valeur vide peut être atteinte en utilisant `" "` ou `' '`. Une valeur vide signifie qu'aucun traitement ne sera activé pour générer un fichier répertoire brut.
- `--identify expression régulière`
  - `-i` *expression régulière* utilisée pour identifier le raccourci et le contenu du répertoire brut avec ce raccourci dans le fichier intermédiaire. La valeur par défaut est :
 

```
^ (\\indexentry)\\([^[^]*)\\(.*)$
```

 pour tous, sauf `splitindex.tlu`. Ça signifie :
    - `^` Recherche du début de la ligne.
    - `(\\indexentry)` Rechercher « `\\indexentry` » et définir avec le groupe 1.
    - `\\[` Rechercher « `[` » et l'ignorer.
    - `([^[^]*)` Rechercher tous les caractères qui ne sont pas « `]` » et les définir avec le groupe 2.
    - `\\]` Rechercher « `]` » et l'ignorer.
    - `(.*)$` Rechercher tous les caractères jusqu'à la fin de la ligne et les définir avec le groupe 3.
 L'*expression régulière* est compatible POSIX 1003.2. Pour `splitindex.tlu`, c'est « `^(\\indexentry)%([^[^]*)%(.*)$` » par défaut.
- `-resultis motif`

**-r *motif*** pour définir les lignes écrites dans le fichier généré du répertoire brut après identification (voir option `-identify`) du motif. Chaque chiffre \$ de motif sera remplacé par le groupe correspondant, par exemple \$1 sera remplacé par le premier groupe (voir `-identify`). La valeur par défaut est « \$1\$3 » pour tous sauf pour `splitindex.tlu` dont la valeur est « %1%3 », ce qui signifie le contenu des groupes 1 et 3.

Si l'expression régulière de l'option `identify` ne correspond pas à une ligne du fichier du répertoire brut, la ligne elle-même sera écrite.

— `-suffixis motif`

**-s *motif*** règle le suffixe sur le nom de fichier du répertoire brut généré au motif identifié (voir option `-identify`), chaque chiffre \$ de motif remplacé par le groupe correspondant, par exemple \$1 sera remplacé par le premier groupe (voir `-identify`). La valeur par défaut « -\$2 » qui s'écrit « -%2 », signifie le caractère « - » suivi par le contenu du groupe 2.

Si l'expression régulière de l'option `identify` ne correspond pas à une ligne du fichier répertoire brut, tous les groupes seront réglés sur `idx`.

— `-verbose`

**-v** augmente la répétition de un. Des moyens redondants indiquent d'une manière plus sûre, à l'utilisateur ce que le programme est en train de faire.

— `-version`

**-V** affiche des informations sur la version du programme et sur la façon de résilier, sans traitement, un fichier répertoire.

Certains binaires compilés à partir du source C ne comprennent pas l'option noms longs (`-makeindex`, `-identify...`), dans ce cas, utiliser l'option noms courts (`-m`, `-i...`). Le premier argument sans option dans la ligne de commande est utilisé comme nom du fichier répertoire intermédiaire à traiter. Tous les arguments qui suivent l'argument « - » sont interprétés comme des arguments sans option. Tous, mais d'abord les arguments sans option, seront transmis au traitement de répertoire. Vous trouverez quelques exemples dans les paragraphes suivants.

### 22.4.11 Utilisation de `splitindex.pl`

Ceci est la mise en œuvre de référence. Prenons un exemple pour démontrer son utilisation. Si vous avez le fichier  $\text{\LaTeX}$  suivant `allabout.tex` :

```
\documentclass{article}
\usepackage[makeindex]{splitidx}
\begin{document}
```

```
Pomme\sindex[fru]{pomme} et orange\sindex[fru]{orange} sont
des fruits\sindex{fruits}.
Salades\sindex[leg]{salade} sont des légumes\sindex{légumes}.
Chats\sindex[ani]{chat} sont des animaux\sindex[idx]{animaux}.
\end{document}
```

cela génère le fichier répertoire intermédiaire `Fileallabout.idx` :

```
\indexentry[fru]{apple}{1}
\indexentry[fru]{orange}{1}
\indexentry{fruits}{1}
\indexentry[veg]{tomato}{1}
\indexentry{vegetables}{1}
\indexentry[ani]{cat}{1}
\indexentry[idx]{animals}{1}
```

Ce fichier ne peut pas être réglé par un traitement de répertoire tel `MakeIndex`. Pour diviser le fichier intermédiaire en plusieurs fichiers bruts puis lancer le processus par défaut, faites l'appel suivant (le `$` est un symbole pour l'invite du shell) :

```
$splitindex.pl allabout.idx
```

Vous pouvez omettre l'extension `.idx` :

```
$splitindex.pl allabout
```

Les deux commandes se traduiront par un fichier `allabout-fru.idx` :

```
\indexentry[fru]{pomme}{1}
\indexentry[fru]{orange}{1}
```

un fichier `allabout-idx.idx`

```
\indexentry{fruits}{1}
\indexentry{vegetaux}{1}
\indexentry{animaux}
```

un fichier `allabout-veg.idx` :

```
\indexentry[veg]{rose}{1}
```

un fichier `allabout-ani.idx` :

```
\indexentry[ani]{chat}{1}
```

Après la génération de ces fichiers, le traitement du répertoire par défaut est appelé en utilisant les lignes de commande :

```
makeindex allabout-fru.idx
```

```
makeindex allabout-idx.odx
makeindex allabout-veg.idx
makeindex allabout-ani.idx
```

Les fichiers de répertoires `allabout-fru.ind`, `allabout-idx.ind`, de même que `allabout-veg.ind` et `allabout-ani.ind` sont créés et peuvent être chargés dans le document en utilisant par exemple `\printindex` du paq `splitidx`. Si vous ne voulez pas appeler `splitindex` pour le traitement du répertoire, utiliser

```
$splitindex.pl -m " " allabout
```

au lieu de la commande `shell` ci-dessus. Vous obtiendrez le même fichier en utilisant `above` (avec une ligne d'entrée et non pas deux comme montré ici) :

```
$splitindex.pl -i '^\\indexentry\\[[^]]*\\)(.*)$' -s '$1'
-r '\\indexentry$2' allabout
```

Pour que `splitindex` appelle `makeindex` avec l'option `foo.ist -s` pour lui faire utiliser le style de fichier `foo.ist` de `MakeIndex`, vous pouvez le faire comme suit :

```
$splitindex.pl allabout - -s foo.ist
```

Comme vous pouvez le voir « - » est utilisé pour empêcher que `splitindex` interprète `-s foo.ist` comme option `--suffixis foo.ist`. Toutes les options de `splitindex` doivent être mises avant « - », mais vous pouvez mettre la première argumentation `allabout` après :

```
$splitindex.pl - allabout -s foo.ist
```

Pour utiliser le traitement de répertoire `xindy` au lieu du traitement par défaut, `MakeIndex`, utiliser cet appel :

```
$splitindex.pl -m xindy allabout
```

Si `xindy` n'est pas dans votre `PATH` standard, vous pouvez définir le chemin complet :

```
$splitindex.pl -m /home/me/bin/xindy allabout
```

Avec la plupart des implémentations de perl, le module perl `Getopt::Long` permet d'ajouter des options après l'argument `non-options` et vous pouvez aussi écrire :

```
$splitindex.pl allabout -m /home/me/bin/xindy
```

pour obtenir le même résultat.



### 22.4.12 Utilisation de `splitindex.jar`

Cette mise en œuvre doit également être portable. Si vous n'utilisez pas Sun Java 1.4.1 ou plus, vous pouvez essayer de recompiler en utilisant la commande `shell` :

```
$javac splitindex.java
```

Cela devrait se traduire par un nouveau `splitindex.class`, mais qui échouera par exemple avec Sun Java 1.3, parce que les expressions régulières nécessaires ne sont pas disponibles dans Sun Java 1.3. L'appel de `splitindex.class` est presque le même que celui indiqué dans le paragraphe 3.11 pour `splitindex.pl` mais vous devez remplacer `index!splitindex.pl` par `splitindex java`. Ainsi, le dernier exemple du paragraphe 3.11 devient :

```
$java splitindex allabout -m /home/me/bin/xindy
```

### 22.4.13 Utilisation de `splitindex.tex`

Le programme `splitindex.tex`  $\text{T}_{\text{E}}\text{X}$  ou  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  ne connaît pas d'option ou d'argument. Le nombre de fichiers qui peuvent être générés est limité au nombre d'écritures libres de titres. Si d'autres lignes que `\indexentry` figurent dans le fichier brut de répertoire, l'usage de `splitindex.tex` produira une erreur. Vous pouvez utiliser `splitindex.tex` interactivement :

```
$tex splitindex
```

ou

```
$latex splitindex
```

Si vous le faites, le nom de fichier du répertoire brut vous sera demandé. Vous devez omettre l'extension `idx` pour répondre à cette question. Vous pouvez bien sûr utiliser `splitindex.tex`, mais pas de manière interactive, par exemple si vous travaillez avec un groupe. Pour ce faire, vous devez définir la macro `\IDX` du nom du premier fichier répertoire sans l'extension `.idx` et ainsi, par exemple le premier du paragraphe 3.11 deviendrait :

```
\$tex \def\IDX{allabout}\input splitindex
```

Vous pouvez également utiliser  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  au lieu de  $\text{T}_{\text{E}}\text{X}$  :

```
\$latex \def\IDX{allabout}\input splitindex
```

La version actuelle de `splitindex.text` ne fait pas appel à tous les procédés de manipulation de répertoires. Mais peut-être qu'une future version sera en mesure de le faire.

### 22.4.14 Fusionner les index

Maintenant, vous devez savoir comment utiliser ensemble les programmes `SplitIndex` et `splitidx` pour sectionner un répertoire. Et comment combiner deux ou plusieurs répertoire en un seul, par exemple placer fruits et légumes dans le même bac<sup>8</sup>, non dans le même répertoire ? Essayer ça :

```
\documentclass{article} % nous utilisons la classe article
\usepackage{splitidx} %...      et le paq splitidx
\makeindex % nous voulons générer un répertoire

% Nous définissons 4 répertoires :

\newindex[General Index]{idx} % nom du 1er répertoire
\newindex[Index Animaux]{ani} % nom du 2ème répertoire
\newindex[Index Fruits et Vegetaux]{fru} % 3ème répertoire
\begin{document}
Pommes\sindex[fru]{pomme} % entrée dans l'index fru
Oranges\sindex[fru]{orange} % entrée dans l'index fru
Fruits\sindex{fruits}. % entrée implicite dans l'index idx
Salades\sindex[leg]{salade} % entrée dans l'index veg
Vegetaux\index{vegetaux}. % entrée implicite dans idx
Chats\sindex[ani]{chat} % entrée dans l'index ani
Animaux\sindex[idx]{animaux}. % entrée explicite dans idx
\printindex* % afficher tous les répertoires
\end{document}
```

puis faire l'appel suivant après le fractionnement du répertoire en utilisant `SplitIndex` :

```
$makeindex allabout-veg.idx allabout-fru.idx
```

Sinon, vous pouvez concaténer `allabout-fru.idx` de `allabout-veg.idx` avant d'exécuter le traitement de répertoire sur `allabout-veg.idx`.

---

8. PAPIRAY précise : je n'ai pas pu m'empêcher, excusez moi.

## 22.5 Mise en œuvre de splitidx

1 \*package

### 22.5.1 Options

La première option est utilisée pour activer la génération de repertoire :

```
2 \DeclareOption{makeindex}{\AtEndOfPackage{\makeindex}}
```

Avec l'option useindex la commande `\index` se comporte comme `\sindex` :

```
3 \DeclareOption{useindex}{%
4   \def\@se@nd@xc@d@{\let\index\sindex}%
5   \AtEndOfPackage{\@se@nd@xc@d@}%
6 }
7 \let\@se@nd@xc@d@\relax
```

Il existe aussi une option pour que `\sindex` ignore l'argument et se comporte comme `\index` :

```
8 \DeclareOption{allatone}{%
9   \PackageWarning{splitidx}{Option ‘allatone obsolète!
10     \MessageBreak
11     Vous devez la remplacer par ‘allintoone}%
12   \ifx\@se@nd@xc@d@\relax\else
13     \PackageInfo{splitidx}{option ‘allatone option de rem-
14       placement ‘useindex}%
15     \let\@se@nd@xc@d@\relax
16   \fi
17   \AtEndOfPackage{%
18     \renewcommand*{\sindex}[1] [] {\index}%
19     \g@addto@macro\makeindex{\renewcommand*{\sindex}[1] []
20       {\index}}%
21   }%
22 }
23 \DeclareOption{allintoone}{%
24   \ifx\@se@nd@xc@d@\relax\else
25     \PackageInfo{splitidx}{option ‘allintoone option de rem-
26       placement ‘useindex}%
27     \let\@se@nd@xc@d@\relax
28   \fi
```

```
25 \AtEndOfPackage{%
26   \renewcommand*\sindex[1] [] {\index}%
27   \g@addto@macro\makeindex{\renewcommand*\sindex[1] []
                                {\index}}%
28 }%
29 }
```

Ne pas développer des arguments de répertoire :

```
30 \newif\if@verbindex\@verbindexfalse
31 \DeclareOption{protected}{\@verbindextrue}
```

Avec l'option `idxcommands` chaque `\newindex` définit aussi une commande de répertoire :

```
32 \newif\if@newidxcmd\@newidxcmdfalse
33 \DeclareOption{idxcommands}{\@newidxcmdtrue}
```

Avec l'option `split`, chaque répertoire utilise son propre fichier :

```
34 \newif\if@splitidx\@splitidxfalse
35 \DeclareOption{split}{\@splitidxtrue}
\
```

Traitement des options :

```
36 \ProcessOptions\relax
```

## 22.5.2 Définition d'une entrée d'index

<pre>\see \seealso \seename \alsoname</pre>
---

Ces quatre macros **standard** sont définies par `makeindex` auquel elles ont été « empruntées ». Non, non, je ne suis pas un mauvais gars, lire `legal.txt`, qui est livré avec `makeidx`.

```
37 \newcommand*\see[2]{\emph{\seename} #1}
38 \providecommand*\seealso[2]{\emph{\alsoname} #1}
39 \providecommand\seename{see}
40 \providecommand*\alsoname{see also}
```

<pre> \index \@wrsindex \@@wrsindex </pre>
--

Fonctionnent de manière similaire à la commande `\index` d'origine, mais en utilisant un répertoire `splittded`, ce qui permet un argument :

```

41 \newcommand*\sindex}[2] []{%
42 }
43 \g@addto@macro\makeindex{%
44 \renewcommand*\sindex}{%
45 \@bsphack\beginngroup
46 \@sanitize
47 \@wrsindex
48 }%
49 \typeout{Using splitted index at \jobname.idx}%
50 \@se@nd@xc@d@
51 }

```

La commande `\@@wrsindex` est utilisée comme agrafe, si elle est définie, pour écrire les entrées de répertoire et peut être utilisée, par exemple, à partir de `hyperref` pour ajouter `hyperpage` à la sélection de la police du numéro de page. Cela fonctionne uniquement avec `encap` | :

```

52 \newcommand*\@wrsindex}[2] []{%
53 \ifx\relax#1\relax
54 \if@splitidx
55 \@wrsindex[idx]{#2}%
56 \else
57 \def\@tempa{#2}%
58 \if@verbinde\@onelevel@sanitize\@tempa\fi
59 \@wrindex{\@tempa}%
60 \fi
61 \else
62 \def\@tempa{#2}%
63 \csname index@#1@hook\endcsname
64 \expandafter\ifx\csname \@wrsindex\endcsname\relax
65 \@@wrsindex{#1}{\@tempa}{\thepage}}%
66 \else
67 \def\@tempb{\@wrsindex{#1}}%
68 \expandafter\@tempb\@tempa||\%

```

```
69 \fi
70 \endgroup
71 \@esphack
72 \fi
73 }
74 \newcommand*{\@@@wrsindex}[2]{%
75 \begingroup
76 \if@splitidx
77 \expandafter\ifx\csname @indexfile@#1\endcsname\relax
78 \PackageError{splitidx}{%
79 Index entry for not existing index%
80 }{%
81 'Youve tried to set an index to index '#1, without
82 defining\MessageBreak
83 that index before using \string\newindex.\MessageBreak
84 This is only allowed, if you are not using package option
85 ''split.%
86 }%
87 \else
88 \expandafter\protected@write\csname @indexfile@#1
89 \endcsname{%
90 \csname index@#1@writehook\endcsname
91 \csname index@#1@writehook@once\endcsname
92 }{%
93 \string\indexentry#2%
94 }%
95 \fi
96 \else
97 \protected@write\@indexfile{%
98 \csname index@#1@writehook\endcsname
99 \csname index@#1@writehook@once\endcsname
100 }{%
101 \string\indexentry[#1]#2%
102 }%
103 \fi
104 }
```

Si hyperref a été chargé par `\begin{document}` et que `hyperref-option hyperindex` n'est pas désactivé, le crochet n'est pas utilisé car indéterminé. Pour le définir :

```

105 \AtBeginDocument{%
106 \begingroup\expandafter\expandafter\expandafter\endgroup
107 \expandafter\ifx\csname ifHy@hyperindex\endcsname\relax
108 \else
109 \expandafter\ifx\csname ifHy@hyperindex\expandafter
    \endcsname
110 \csname iftrue\endcsname
111 \@ifundefined{@@wrsindex}{%
112 \def\@@wrsindex#1#2|#3|#4\{\%
113 \ifx\#3\%
114 \@@@wrsindex{#1}{{#2|hyperpage}{\thepage}}%
115 \else
116 \def\Hy@temp@A{#3}%
117 \ifx\Hy@temp@A\HyInd@ParenLeft
118 \@@@wrsindex{#1}{{#2|#3hyperpage}{\thepage}}%
119 \else
120 \ifx\Hy@temp@A\HyInd@ParenRight
121 \@@@wrsindex{#1}{{#2|#3hyperpage}{\thepage}}%
122 \else
123 \@@@wrsindex{#1}{{#2|#3}{\thepage}}%
124 \fi
125 \fi
126 \fi
127 }%
128 }{}%
129 \fi
130 \fi
131 }

```

<code>\AtWriteToIndex</code>
------------------------------

Ajouter des commandes pour l'écriture de l'agrafe :

```

132 \newcommand*\AtWriteToIndex[1]{%
133 \expandafter\ifx\csname index@#1@writehook\endcsname
    \relax
134 \expandafter\let\csname index@#1@writehook\endcsname

```

```

\@empty
135 \fi
136 \expandafter\g@addto@macro\csname index@#1@writehook
\endcsname
137 }

```

`\AtNextWriteToIndex`

Analogue à `\AtWriteToIndex` une seule fois :

```

138 \newcommand*\AtNextWriteToIndex}[1]{%
139 \expandafter\ifx\csname index@#1@writehook@once
\endcsname\relax
140 \expandafter\gdef\csname index@#1@writehook@once
\endcsname{%
141 \expandafter\global\expandafter\let\expandafter
142 \csname index@#1@writehook@once\endcsname\relax
143 }%
144 \fi
145 \expandafter\g@addto@macro\csname index@#1@writehook
@once\endcsname
146 }

```

### 22.5.3 Édition d'un ou de plusieurs index

`\printindex \printindex*`

Ceci est utilisé pour éditer un répertoire de façon normale. Dans la plupart des cas, il utilise l'environnement `theindex`, mais il n'en a pas besoin :

```
147 \newcommand*\printindex}{%
```

La commande peut être appelée dans la version étoilée qui édite les répertoires définis. C'est la même que `printindex` :

```

148 \@ifstar {%
149 \begingroup
150 \let\printindex@@endhook=\printindex@endhook
151 \let\printindex@endhook=\relax
152 \printindex%
153 \csname printindex@@endhook\endcsname
154 \endgroup

```



```
155 }{%
```

Elle peut également être appelée avec les arguments d'édition des répertoires :

```
156 \@ifnextchar [\@printindex %] brace check comment
```

ou appelée sans aucun paramètre et donc identique au `paq makeidx` :

```
157 {%
158   \@input@{\jobname.ind}%
159   \csname printindex@endhook\endcsname
160 }%
161 }%
162 }
```

`\@printindex`

Ceci est utilisé pour éditer un des répertoires. L'argument en option (qui est ici obligatoire) est le raccourci du répertoire :

```
163 \newcommand*\@printindex[1]{}
164 \def\@printindex[#1]{%
```

Il peut y avoir un argument qui est le nom du répertoire. Sinon, la valeur du titre par défaut `\index@raccourci@nom` est utilisée :

```
165 \@ifnextchar [%
166   {\@printindex[#1]}%
167   {\@printindex[#1][\csname index@#1@name\endcsname]}%
168 }
```

`\@@printindex`

Nous utilisons l'environnement par défaut pour éditer un des répertoires, mais nous redéfinissons `\indexname` avec le titre de l'index désiré, `\indexshortcut` avec le raccourci de ce même index et `\index@préambule` avec le préambule du répertoire souhaité. Nous le faisons dans le groupe qui reste donc local :

```
169 \newcommand*\@@printindex[1]{}
170 \def\@@printindex[#1][#2]{%
171   \begingroup
172   \edef\indexshortcut{#1}%
173   \def\indexname{#2}%
```

La documentation de la bibliothèque `tcolorbox` utilise `\kvtcb@text@index` au lieu de `\indexname` et nous avons également redéfini cette commande :

```

174 \def\kvtcb@text@index{#2}%
175 \let@index@preamble\relax
176 \expandafter\let\expandafter@index@preamble
177 \csname index@\indexshortcut @preamble\endcsname
178 \if@splitidx
179   \def\@tempa{idx}\def\@tempb{#1}%
180   \ifx\@tempa\@tempb\let\@indexsuffix\@gobble\fi
181 \fi
182 \@input@{\jobname\@indexsuffix{#1}.ind}%
183 \endgroup
184 \csname printindex@endhook\endcsname
185 }

```

#### `\@indexsuffix`

Cela génère le suffixe à partir du raccourci. Si vous en avez besoin, vous pouvez redéfinir cette fonction. J'utilise ici une astuce qui consiste à définir la macro avec ses propres catcodes, en local mais pas en global. Vous pouvez également utiliser `\@firstofone` au lieu de `\lowercase` :

```

186 \begingroup
187 \catcode\ -12
188 \lowercase{\endgroup
189 \newcommand*\@indexsuffix}[1]{-#1}%
190 }%

```

#### `\printindices`

Ceci est utilisé pour éditer tous les répertoires dans l'ordre de leur définition et de leur titre par défaut.

Si la liste est vide, elle se comporte comme `\printindex` sans étoiles avec des arguments :

```

191 \newcommand*\printindex{%
192 \ifx\@indices\@empty
193 \printindex
194 \else
195 \begingroup
196   \@for\@tempa:=\@indices\do{%
197     \expandafter\printindex\expandafter[\@tempa]%
198   }%
199 \endgroup

```

```
200 \fi
201 }
```

```
\newindex
```

La définition d'un répertoire nouveau comporte un argument obligatoire, le nom raccourci pour ce répertoire, et un argument, le nom de ce répertoire. Si vous ne passez pas l'argument, le raccourci est utilisé comme nom par défaut. si l'index. La définition est globale :

```
202 \newcommand*{\newindex}[2] [\relax]{%
203 \@ifundefined{index@#2@name}{%
204 \if@verbindindex
205 \expandafter\gdef\csname index@#2@hook
206 \endcsname{%
207 \@onelevel@sanitize\@tempa
208 }%
209 \else
210 \expandafter\gdef\csname index@#2@hook
211 \endcsname{}}%
212 \fi
213 \ifx\@indices\@empty
214 \xdef\@indices{#2}%
215 \else
216 \xdef\@indices{\@indices,#2}%
217 \fi
218 \ifx \relax#1
219 \expandafter\xdef\csname index@#2@name
220 \endcsname{#2}%
221 \else
222 \expandafter\xdef\csname index@#2@name
223 \endcsname{#1}%
224 \fi
225 \if@newidxcmd
226 \expandafter\newcommand\expandafter*\csname #2
227 \endcsname{%
228 \expandafter\gdef\csname #2\endcsname{%
229 \sindex[#2]%
230 }%
231 \fi
```

```

227 \if@splitidx
228   \def\@tempa{#2}\def\@tempb{idx}%
229   \ifx\@tempa\@tempb
230     \global\let\@indexfile@idx=\@indexfile
231   \else
232     \expandafter\newwrite\csname @indexfile@#2
\endcsname
233     \expandafter\immediate\expandafter\openout
234     \csname @indexfile@#2\endcsname=
\jobname-#2.idx
235   \fi
236 \fi
237 }{%

```

Si le répertoire est déjà défini, une erreur se produit :

```

238 \PackageError{splitidx}{%
239   index ‘’#2 already defined%
240 }{%
241   You have already defined an index with shortcut ‘’#2.
\MessageBreak
242   You ’cant define a new index with the same shortcut.
If ’youll continue
243   \MessageBreak
244   The new definition will be ignored.%
245 }%
246 }%
247 }
248 \if@splitidx
249 \@onlypreamble\newindex
250 \fi

```

`\newprotectedindex`

Comme avec `\newindex`, le répertoire est toujours défini avec des arguments protégés :

```

251 \newcommand*{\newprotectedindex}[2][\relax]{%
252 \begingroup\@verbindextrue\newindex[#{1}]{#2}\endgroup
253 }

```

`\@indices` Cette macro stocke une liste des raccourcis de répertoires qui sont nécessaires par exemple pour `\printindex` ou pour `\newindex` :

```
254 \newcommand*\@indices{}
255 \gdef\@indices{}
```

`\extendtheindex`

Permet d'étendre `theindex` avec des macros appelées avant de commencer, après le démarrage, avant d'arrêter et après l'arrêt du répertoire. Cela peut être utilisé pour modifier son comportement. Un autre changement est fait, qui peut être utile avant que le répertoire `\index@preamble` ne soit réglé sur `\index@raccourci@preamble` :

```
256 \newcommand{\extendtheindex}[4]{%
257 \begingroup\expandafter\expandafter\expandafter\endgroup
258 \expandafter\ifx\csname splitindex@theindex\endcsname
   \relax
259 \let\splitindex@theindex=\theindex
260 \let\endsplitindex@theindex=\endtheindex
261 \fi
262 \renewcommand*\theindex{%
263 #1\splitindex@theindex #2%
264 }%
265 \renewcommand*\endtheindex{%
266 #3\endsplitindex@theindex #4%
267 }%
268 }
```

`\setindexpreamble`

Définir un des préambules *splitted* du répertoire ou celui d'origine :

```
269 \newcommand{\splitindex@setip}{}
270 \let\splitindex@setip\setindexpreamble
271 \let\setindexpreamble\relax
272 \newcommand{\setindexpreamble}[2][ ]{%
273 \ifx \relax#1\relax
274 \begingroup\expandafter\expandafter\expandafter
   \endgroup
275 \expandafter\ifx\csname splitindex@setip\endcsname
   \relax
276 \@namedef{index@preamble}{#2}%
```

```

277 \else
278   \splitindex@setip{#2}%
279 \fi
280 \else
281 \@namedef{index@#1@preamble}{#2}%
282 \fi
283 }

```

\useindexpreamble

Utiliser le préambule du répertoire s'il existe et s'il n'est pas vide et l'ajouter en option après des informations supplémentaires :

```

284 \newcommand{\useindexpreamble}[1] [] {%
285 \begingroup\expandafter\expandafter\expandafter
   \endgroup
286 \expandafter\ifx\csname index@preamble\endcsname\relax
   \else
287 \ifx\index@preamble\@empty\else
288 \index@preamble #1%
289 \fi
290 \fi
291 }

```

\printsubindex \printsubindex\*

Fonctionne comme \printindex mais, avant de les niveler vers le bas, modifie certaines macros de rubriques lors de la génération du répertoire :

```

292 \newcommand*{\printsubindex}{%
293 \begingroup
294 \begingroup\expandafter\expandafter\expandafter
   \endgroup
295 \expandafter\ifx\csname chapter\endcsname\relax
296 \let\section\subsection
297 \begingroup\expandafter\expandafter\expandafter
   \endgroup
298 \expandafter\ifx\csname addsec\endcsname\relax\else
299 \def\addsec{\setcounter{secnumdepth}{0}\subsection}%
300 \fi
301 \else
302 \let\chapter\section

```

```

303 \def\@makeschapterhead{\section*}
304 \let\@makechapterhead\section
305 \begingroup\expandafter\expandafter\expandafter
   \endgroup
306 \expandafter\ifx\csname addchap\endcsname\relax\else
307 \let\addchap\addsec
308 \fi
309 \fi

```

En outre, `\`, `\twocolumn` et `\clearpage` doivent être désactivés. Les macros `\onecolumn` et `\twocolumn` ne peuvent pas être `\relax` parce qu'elle ont un argument optionnel qui doit être utilisé :

```

310 \let\onecolumn\@firsttofone
311 \let\twocolumn\@firsttofone
312 \let\clearpage\relax
313 \let\cleardoublepage\relax

```

et le mécanisme de marque doit également en utiliser un vers le bas :

```

314 \def\markboth{\expandafter\markright\@gobble}%
315 \ifx\@mkboth\@gobble\else\let\@mkboth\markboth\fi
And the page style 'shouldnt change too:
316 \let\thispagestyle\@gobble

```

Maintenant, `\printindex` permet d'appliquer toutes ces fonctions :

```

317 \let\printindex@endhook=\endgroup
318 \printindex
319 }

```

`\@firsttofone`

Lire l'argument optionnel et l'exécuter :

```

320 \providecommand{\@firsttofone}[1] [] {#1}
321 /package

```





# Références

Comme je l'ai indiqué en début d'ouvrage, j'ai démarré ce travail, en mai 2014, sur la base de la version de l'époque qui a évolué le 16 avril, le 15 septembre, le 3 octobre 2015, avant de devenir la v3.20 en date du 10 mai 2016, la v3.21 le 14 juin puis la v3.22 le 2 janvier 2017, enfin la v3.23 le 13 avril 2017 et chaque fois, pour coller à la réalité, je me suis adapté en intégrant ces modifications.

KOMA-Script de Markus KOHM en est bien entendu le noyau, mais j'aborde un système d'encodage qui m'était inconnu,  $\LaTeX$ , et je me suis inspiré, pour le découvrir à travers  $X\LaTeX$ , des ouvrages suivants :

Vincent LOZANO, *Tout ce que vous avez toujours voulu savoir sur  $\LaTeX$  sans jamais oser le demander*, Framabook/Framasoft, 2013, <https://framabook.org/tout-sur-latex/>

Noel-Arnaud MAGUS, *Rédigez des documents de qualité avec  $\LaTeX$* , Éditions **le site du Zéro**, [www.siteduzero.com](http://www.siteduzero.com).

Maïeul ROUQUETTE,  *$X\LaTeX$  appliqué aux sciences humaines*, Éditions **Atramenta**, 2012, [www.atramenta.net](http://www.atramenta.net).

Sébastien COMBÉFIS,  *$\LaTeX$  HowTo – Le Guide Pratique*, UKO Publication, 2010, <http://latex-howto.be>.

Céline CHEVALIER et *al.*,  *$\LaTeX$  pour l' impatient* H&K Minimax, 2009, [http://www.h-k.fr/liens/tp/latex\\_pour\\_l\\_impatient.html](http://www.h-k.fr/liens/tp/latex_pour_l_impatient.html).

Sans oublier :

- le wiki des fiches à BÉBERT : <http://lesfichesabebert.fr/>
- la FAQ latex de Fabien Torre : <http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>
- le site UKO de nos amis belges, créé par Sébastien COMBÉFIS : <http://www.ukonline.be/programmation/latex/index.php>

Cette liste n'est pas exhaustive et ne mentionne que les quelques ouvrages et sites parcourus le plus fréquemment.

Quant au bon usage de la langue française, j'ai toujours, à portée de main,

- *Orthotypographie* de Jean-Pierre LACROUX, les volumes un et deux de l'édition originale ;
- *Le Ramat européen de la Typographie* d'Aurel RAMAT et Romain MULLER aux éditions **De Champlain** ([editions-dechamplain@laposte.net](mailto:editions-dechamplain@laposte.net)) ;
- le *Petit guide de Typographie* d'Éric MARTINI paru aux Éditions **Glyphe** ([www.editions-glyphe.com](http://www.editions-glyphe.com)).

que je consulte régulièrement. En cas de doute, il me reste trois références françaises solides :

- le *Lexique des règles typographiques en usage à l'imprimerie nationale* édition 2002,
- le site « *Renouvo* », réseau pour la nouvelle orthographe du français qui diffuse l'information sur les rectifications orthographiques proposées et recommandées par les instances francophones compétentes (entre autres, l'Académie française et « les » Conseil supérieur de la langue française), <http://www.renouvo.org/index.php>
- le forum de l'*ABC de la langue française* <http://www.languefrancaise.net/forum/index.php>.

et une référence éclairée de nos amis canadiens francophones :

<http://www.oqlf.gouv.qc.ca/accueil.aspx>

Vous trouverez ci-après, sous le titre bibliographie, un ensemble complet des renvois appelés dans le texte. Dans de nombreux cas, il existe des documents ou des répertoires entiers disponibles sur Internet. Alors une URL est citée. S'il est fait référence à un paq L<sup>A</sup>T<sub>E</sub>X, cette dernière est généralement sous la forme « CTAN : // Lien » qui signifie que l'archive est sur un serveur CTAN ou sur un miroir. Vous pouvez remplacer le préfixe, par `http://mirror.ctan.org/`.

Pour les paqs L<sup>A</sup>T<sub>E</sub>X, la version utilisée pour effectuer l'essai devrait être notée avec une référence également spécifiée dans le texte. Mais dans le cas de certains paqs il s'agit d'un jeu de devinettes pour trouver une approche cohérente du numéro de version et de la date de sortie, et de plus, la version spécifiée n'est pas toujours la dernière version disponible.

Si vous chargez un paq pour l'installer, vous devez toujours essayer d'abord la version actuelle, mais avant de le télécharger, vous assurer qu'il n'est pas déjà sur votre ordinateur.

---

Enfin, je navigue également, au hasard de mes hésitations (merci Mozilla), sur de nombreux sites que je ne peux pas, malheureusement, tous citer ici.

Que leurs auteurs et animateurs ne m'en tiennent pas rigueur.



# Bibliographie

- [Ame02] AMEMATSOC. *le guide de l'utilisateur pour le pack amsmath*. American Mathematical Society, Février 2002. URL : CTAN : //macros/latex/required/amslatex/math/.
- [BB13] Johannes BRAAMS et Javier BEZOS. *Babel*. Décembre 2013. URL : CTAN://macros/latex/required/babel/.
- [Bra+05] Johannes BRAAMS, David CARLISLE et al. *The LaTeX Source*. Décembre 2005.
- [Car04] David CARLISLE. *The longtable pack*. Février 2004. URL : CTAN : //macros/latex/required/tools/.
- [Car05] David CARLISLE. *Packages in the graphics bundle*. Novembre 2005. URL : CTAN://macros/latex/required/graphics/.
- [Car99a] David CARLISLE. *The tabularx package*. Janvier 1999. URL : CTAN://macros/latex/required/tools/.
- [Car99b] David CARLISLE. *The keyval package*. Mars 1999. URL : CTAN : //macros/latex/required/graphics/.
- [Che11] Florent CHERVET. *tabu and longtabu*. Février 2011. URL : CTAN://macros/latex/contrib/tabu/.
- [Dal10] Patrick W. DALY. *Natural sciences citations and references*. Septembre 2010. URL : CTAN : //macros/latex/contrib/natbib/.

- [Dan+12] Marco DANIEL et al. *LaTeX-Kurzbeschreibung*. Juli 2012. URL : CTAN://info/lshort/german/.
- [DUD96] DUDEN. *Die deutsche Rechtschreibung*. DUDENVERLAG, Mannheim, 21. Auflage, 1996.
- [Fai11] Robin FAIRBAIRNS. *footMisc – a portmanteau package for...* Juin 2011. URL : CTAN://macros/latex/contrib/footMisc/.
- [Gau07] Bernard GAULLE. *Les distributions de fichiers de francisation*. Mai 2007. URL : CTAN://language/french/.
- [Gre12] Enrico GREGORIO. *The xpatch package, extending etoolbox...* Oktober 2012. URL : CTAN://macros/latex/contrib/xpatch/.
- [KDP] KDP. KOMA-Script Homepage. URL : <http://www.komascript.de>.
- [Keh97] Roger KEHR. *XINDY, A Flexible Indexing System*. 1997.
- [Ker07] Uwe KERN Dr. *Extending LaTeX's color facilities : the xcolor...* Janvier 2007. URL : CTAN://macros/latex/contrib/xcolor/.
- [Kie10] Axel KIELHORN. *adrconv*. Avril 2010. URL : CTAN://macros/latex/contrib/adrconv/.
- [Knu90] Donald KNUTH E. *The TeXbook, Band A der Reihe Computers...* 1990.
- [Koh02] Markus KOHM. *Satzspiegelkonstruktionen im Vergleich...* DANTE e. V., 2002.
- [Koh03] Markus KOHM. *Moderne Briefe mit KOMA-Script...* DANTE e. V., 2003.
- [Koh14a] Markus KOHM. *KOMA-Script*. Sous la dir. de Berlin EDITIONS DANTE. LEHMANN'S MEDIA. 2014.
- [Koh14b] Markus KOHM. *Creating more than one index using splitidx...* Avril 2014. URL : CTAN://macros/latex/contrib/splitindex.
- [Lam87] Leslie LAMPORT. *MakeIndex : An Index Processor For LaTeX*. Février 1987. URL : CTAN://indexing/makeindex/doc/makeindex.pdf.

- 
- [Lap08] Olga LAPKO. *The floatrow pack*. Août 2008. URL : CTAN://macros/latex/contrib/floatrow/.
- [Leh11] Philipp LEHMAN. *The etoolbox pack*. Janvier 2011. URL : CTAN://macros/latex/contrib/etoolbox/.
- [Lin01] Anselm LINGNAU. *An improved environment for floats*. nov. 2001. URL : CTAN://macros/latex/contrib/float/.
- [Mit11] Frank MITTELBACH. *An environment for multicolumn output*. Juin 2011. URL : CTAN://macros/latex/required/tools/.
- [Nie15] Rolf NIEPRASCHK. *The eso-pic package*. Juli 2015. URL : CTAN://macros/latex/contrib/eso-pic/.
- [Obe07] Heiko OBERDIEK. *The selinput pack*. September 2007. URL : CTAN://macros/latex/contrib/oberdiek.
- [Obe09] Heiko OBERDIEK. *The picture package*. Oktober 2009. URL : CTAN://macros/latex/contrib/oberdiek/picture.dtx.
- [Oos04] Piet van OOSTRUM. *Page layout in LaTeX*. Mars 2004. URL : CTAN://macros/latex/contrib/fancyhdr/.
- [Pac] Jean-Marie PACQUET. *KomaLetter2; Example by French style*. Sous la dir. de WIKI. URL : <http://wiki.lyx.org/Examples/KomaLetter2#toc6>.
- [Rai98a] Bernd RAICHLE. *german pack*. Juillet 1998. URL : CTAN://language/german/.
- [Rai98b] Bernd RAICHLE. *ngerman pack*. Juillet 1998. URL : CTAN://language/german/.
- [Sch09] Martin SCHRÖDER. *The ragged2e pack*. Juni 2009. URL : CTAN://macros/latex/contrib/ms/.
- [Sch13] R. SCHLICHT. *The microtype pack : An interface to the micro...* Mai 2013. URL : CTAN://macros/latex/contrib/microtype.
- [Tea05a] LaTeX Project TEAM. *LaTeX font selection*. Novembre 2005. URL : CTAN://macros/latex/doc/fntguide.pdf.
- [Tea05b] LaTeX Project TEAM. *LaTeX for authors*. Novembre 2005. URL : CTAN://macros/latex/doc/usrguide.pdf.

- [Tea06] LaTeX Project TEAM. *LaTeX for class and pack writers*. Février 2006. URL : [CTAN://macros/latex/doc/clsguide.pdf](http://CTAN://macros/latex/doc/clsguide.pdf).
- [Tea98] The NTS TEAM. *The  $\varepsilon$ -TeX manual*. Februar 1998. URL : [CTAN://systems/e-tex/v2/doc/etex\\_man.pdf](http://CTAN://systems/e-tex/v2/doc/etex_man.pdf).
- [TF11] Geoffrey TOBIN et Robin FAIRBAIRNS. *setspace LaTeX package*. Décembre 2011. URL : [CTAN://macros/latex/contrib/setspace/](http://CTAN://macros/latex/contrib/setspace/).
- [Tsc60] Jan TSCHICHOLD. *Erfreuliche Drucksachen durch gute Typo...* Sous la dir. de Ravensburger Buchverlag Otto MAIER. 1960.
- [Tsc87] Jan TSCHICHOLD. *Ausgewählte Aufsätze über Fragen der gest...* Sous la dir. de 2. Auflage BIRKHÄUSER VERLAG Basel. 1987.
- [Ume10] Hideo UMEKI. *The geometry package*. Septembre 2010. URL : [CTAN://macros/latex/contrib/geometry/](http://CTAN://macros/latex/contrib/geometry/).
- [WF00] Hans-Peter WILLBERG et Friedrich FORSSMAN. *Erste Hilfe...* Sous la dir. de Mainz VERLAG HERMANN SCHMIDT. 2000.
- [Wik] WIKI. *Deutsche TeX-FAQ*. Sous la dir. de WIKI. URL : <http://projekte.dante.de/DanteFAQ/WebHome>.



# Liste des figures

1.1	Empagement, marges et empage . . . . .	4
1.2	Construction classique d'une grille avec DIV par neuf . . . . .	8
2.1	Paramètres de la présentation des notes . . . . .	80
2.2	Utilisation de <code>\captionaboveof</code> dans un environnement flottant	128
2.3	Légende de la figure alignée en bas sur le côté . . . . .	131
2.4	Légende de la figure sur le côté alignée au milieu . . . . .	132
2.5	Légende sur le côté alignée en haut . . . . .	132
2.6	Exemple : légende avec prépositionnement . . . . .	136
2.7	Exemple : légende avec retrait négatif partiel . . . . .	136
2.8	Exemple : légende avec indentation fidèle à la mise en page . . .	136
2.9	Exemple : légende avec entrée dans la deuxième rangée . . . . .	136
3.1	Exemple de construction d'un document avec un nombre quel- conque de lettres . . . . .	163
3.2	Exemple de construction générale d'une seule lettre . . . . .	163
3.3	Lettre minimum avec uniquement adresse et ouverture . . . . .	168
3.4	Lettre avec adresse, ouverture, titre, texte et salutations . . . . .	169
3.5	Résultat d'une lettre avec adresse, ouverture, titre, texte, ferme- ture et post-scriptum (date et marques de pliage sont par dé- faut des lettres DIN) . . . . .	170
3.6	Lettre avec adresse, ouverture, texte, fermeture, post-scriptum, et liste de distribution . . . . .	172
3.7	Lettre avec adresse, ouverture, texte, fermeture, post-scriptum, pièce jointe, et liste de distribution . . . . .	173
3.8	Lettre avec adresse, titre, texte, fermeture de phrase, Postscript, installation, distribution et gros caractère de police . . . . .	178
3.9	Affichage schématique du papier à lettres avec les principales commandes et variables pour les éléments décrits . . . . .	181

3.10	Lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation . . . . .	182
3.11	Lettre de l'expéditeur, adresse, ligne de partage, titre, texte, phrase de fermeture, signature, post-scriptum, pièce jointe, distribution et marque de poinçon . . . . .	187
3.12	Lettre de l'expéditeur élargi, ligne de partage, adresse, ouverture, texte, phrase de fermeture, signature, post-scriptum, pièce jointe, liste de diffusion et marque de poinçon . . . . .	191
3.13	Lettre de l'expéditeur élargi, ligne de partage, adresse, ouverture, texte, fermeture de la phrase, signature, post-scriptum, liste de diffusion et marque de poinçon . . . . .	192
3.14	expéditeur aligné à gauche avec logo à droite . . . . .	194
3.15	expéditeur centré avec logo en dessous à droite, et expéditeur aligné à droite avec logo à gauche . . . . .	194
3.16	Lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation . . . . .	202
3.17	Lettre avec logo, adresse, extension de l'expéditeur, lieu, date, sujet, titre, texte, fermeture, signature, post-scriptum, annexe, distribution et marque de pliage . . . . .	207
3.18	Lettre avec logo, adresse, extension de l'expéditeur, lieu, date, marque de respect, sujet, titre, texte, fermeture, signature, post-scriptum, annexe, distribution et marque de pliage . . . . .	210
3.19	Résultat d'une lettre avec ouverture, titre, texte, fermeture, post-scriptum, aménagements, liste de diffusion et marque de perforation (la date est pré-réglée pour DIN-lettres) . . . . .	232
3.20	Lettre avec ouverture, logo, adresse, date, sujet, titre, texte, fermeture, signature, post-scriptum, aménagements, liste de diffusion et marque de perforation avec un fichier <code>.ico</code> . . . . .	236
4.1	Sémantique des commandes définissant l'entête d'un style de page <code>recto-verso</code> . . . . .	254
4.2	Sémantique des commandes définissant le contenu du pied d'un style de page <code>recto-verso</code> . . . . .	257
14.1	Illustration de certains attributs de style d'entrée de répertoire de <code>dottedtocline</code> . . . . .	438
14.2	Illustration de certains attributs de style d'entrée de répertoire de <code>largetocline</code> . . . . .	439

14.3	Illustration de certains attributs de style d'entrée de répertoire de <code>tocline</code> . . . . .	441
14.4	Illustration de certains attributs de <code>undottedtocline</code> comme style d'entrée de répertoire pour un titre de chapitre . . . . .	441
17.1	Éléments des trois parties d'un style de page double face avec les instructions <code>deftriplepagestyle</code> , <code>newtriplepagestyle</code> , <code>providetriplepagestyle</code> et <code>renewtriplepagestyle</code> . . . . .	518
21.1	Représentation schématique des pseudo-longueurs les plus importantes pour entête . . . . .	579



# Liste des tables

1.1	Ratios de principes utilisés pour la mise en page . . . . .	5
1.2	Valeurs de mise en page en fonction de DIV pour A4 . . . . .	15
1.3	Paramètres prédéfinis de DIV pour A4 . . . . .	15
1.4	Valeurs symboliques possibles pour DIV . . . . .	18
1.5	Argument BCOR symbolique possible . . . . .	19
1.6	Valeurs standard pour de simples interrupteurs . . . . .	20
1.7	Pilotes de sortie pour l'option pagesize=output . . . . .	31
2.1	Correspondance entre classes <code>standard</code> et <code>KOMA-Script</code> . . . . .	37
2.2	Éléments modifiables dans <code>scrbook</code> , <code>scrreprt</code> ... . . . . .	45
2.3	Préréglages de la police pour les éléments du titre . . . . .	53
2.4	Taille de la police et alignement des éléments de la page de titre principal lors de l'utilisation de <code>maketitle</code> . . . . .	54
2.5	Valeurs possibles de l'option <code>toc</code> pour la table des matières . . . . .	59
2.6	Paramètres par défaut de la police pour la table des matières . . . . .	63
2.7	Valeurs possibles de <code>parskip</code> pour le marquage des paragraphes . . . . .	65
2.8	Paramètres par défaut de la police des éléments style de page . . . . .	69
2.9	Macros pour définir le style de pages spéciales . . . . .	71
2.10	Styles de numérotation disponible pour numéros de page. . . . .	73
2.11	Valeurs possibles de l'option <code>footnote</code> . . . . .	77
2.12	Valeurs possibles des intercalaires avec des pages blanches . . . . .	83
2.13	Valeurs possibles de l'option <code>rubriques</code> pour régler les entêtes . . . . .	86
2.14	Valeurs possibles d'options pour le point final de numéro d'unité . . . . .	89
2.15	Paramètres par défaut de la police pour <code>scrbook</code> et <code>scrreprt</code> . . . . .	95
2.16	Police pour maxime . . . . .	107
2.17	Valeurs possibles de mise en forme pour légendes de flottants . . . . .	123
2.18	Exemple de table discutable . . . . .	126
2.19	Meilleur exemple de table . . . . .	127
2.20	Masse rectangulaire . . . . .	128
2.21	Valeurs possibles de l'option <code>listof</code> les répertoires des flottants . . . . .	140

2.22	Valeurs prédéfinies pour régler l'option bibliographie . . . . .	147
2.23	Valeurs possibles de l'option pour le réglage de l'index . . . . .	149
3.1	Liste des variables compatibles avec <code>scrlttr2</code> . . . . .	152
3.2	Liste alphabétique des éléments dont la police peut être modifiée avec <code>setkomafont</code> et <code>addtokomafont</code> . . . . .	176
3.3	Valeurs combinables pour la configuration des marques de pliage avec l'option <code>foldmarks</code> . . . . .	179
3.4	Valeurs possibles de l'option <code>fromalign</code> pour définir la position de l'adresse dans l'entête de <code>scrlttr2</code> . . . . .	185
3.5	Valeurs possibles de l'option <code>fromrule</code> pour la position d'une ligne horizontale dans <code>fromaddress</code> avec <code>scrlttr2</code> . . . . .	186
3.6	Étiquettes prédéfinies des variables pour les informations de l'expéditeur dans l'entête . . . . .	188
3.7	Étiquettes et contenu du séparateur pour informations d'expéditeur prédéfinies dans l'entête sans option <code>symbolicnumbers</code> . . . . .	189
3.8	Préférences pour la police des éléments de la fenêtre d'adresse . . . . .	197
3.9	Valeurs possibles de l'option <code>addrfield</code> pour le destinataire . . . . .	198
3.10	Valeurs possibles de l'option <code>priority</code> d'adresse internationale . . . . .	199
3.11	Valeurs possibles de l'option <code>locfield</code> pour le réglage de la largeur du champ de l'expéditeur avec <code>scrlttr2</code> . . . . .	200
3.12	Valeur possible de l'option <code>refline</code> pour le réglage de la largeur des champs de la ligne de référence avec <code>scrlttr2</code> . . . . .	203
3.13	Descriptions prédéfinies des variables des champs de référence en utilisant des macros en fonction du langage courant . . . . .	204
3.14	Préférences pour la police des éléments d'occupation de ligne . . . . .	206
3.15	Désignations prédéfinies de la variable pour le sujet . . . . .	208
3.16	Valeurs possibles pour l'option avec <code>scrlttr2</code> sous réserve de mise en place et formatage du sujet . . . . .	209
3.17	Valeurs possibles de l'option <code>pagenumber</code> pour positionner la page style des rubriques <code>myheadings</code> et <code>plain</code> . . . . .	218
3.18	Fichiers <code>.ico</code> prédéfinis . . . . .	236
4.1	Éléments de <code>scrlayer-scrpage</code> dont le type de style peut être changé avec <code>KOMA-Script</code> et paramètres par défaut si ceux-ci ne sont pas définis avant le chargement de <code>scrlayer-scrpage</code> . . . . .	252
4.2	Valeurs possibles de l'option <code>markcase</code> pour le choix des minuscules/majuscules des entêtes de colonnes automatiques . . . . .	266

4.3	Valeurs symboliques autorisées pour les options de largeur d'en-tête et de pied de page . . . . .	272
7.1	Quatre options si la commande d'accès <code>\FreeIV</code> a été utilisée avec une entrée <code>\adrentry</code> . . . . .	286
9.1	Aperçu des fonctionnalités étendues en option de <code>scrextend</code> . . . . .	292
10.1	Liste des arguments optionnels . . . . .	320
10.2	Références possibles pour <code>S1</code> , <code>alinéa1</code> , <code>composition1</code> . . . . .	322
10.3	Caractéristiques possibles pour les arguments optionnels des commandes <code>\Clause</code> et <code>\SubClause</code> . . . . .	360
10.4	Valeurs possibles de l'option <code>clausemark</code> pour générer des titres de colonnes par paragraphes . . . . .	362
10.5	Valeurs possibles de l'option <code>ref</code> pour régler le format de <code>\ref</code> , <code>\refPar</code> et <code>\refSentence</code> . . . . .	367
10.6	Exemple de tâches de l'option <code>ref</code> avec instructions indépendantes pour les références croisées . . . . .	368
10.7	Signification et préférences pour les termes dépendants de la langue qui ne sont pas encore définis . . . . .	369
11.1	Ensemble de termes habituels dépendant du paq de langue . . . . .	400
14.1	Attributs pour le répertoire prédéfini liste de styles de <code>tocbasic</code> . . . . .	442
14.2	Comparaison des environnements échantillons et figures . . . . .	456
14.3	Options pour la commande <code>\DeclareNewTOC</code> . . . . .	457
16.1	Options pour la définition des couches de page avec la description de l'attribut correspondant . . . . .	477
16.2	Options entre crochets pour les styles de page (dans l'ordre de leur exécution) . . . . .	490
17.1	Noms des niveaux du style de page définis par <code>scrlayer-scrpage</code> . . . . .	520
18.1	Réglages possibles pour la déclaration de colonnes de notes . . . . .	528
20.1	Styles de titre et valeurs des propriétés déclarées dans le dispositif de commandes . . . . .	555
20.2	Clés supplémentaires et valeurs des propriétés dans la déclaration de commandes du style <code>section</code> . . . . .	557

20.3	Clés supplémentaires et valeurs des propriétés dans la répartition de la configuration des commandes du style <code>chapter</code> . . . . .	558
20.4	Clés supplémentaires et valeurs des propriétés dans la répartition de la configuration des commandes du style <code>part</code> . . . . .	559
20.5	Préférences pour les rubriques de <code>scrbook</code> et <code>scrreprt</code> . . . . .	560
20.6	Préférences pour les entêtes de chapitre de <code>scrbook</code> et <code>scrreprt</code> avec l'option <code>headings</code> . . . . .	561
21.1	Pseudo-longueurs utilisées par la classe <code>scr1ttr2</code> . . . . .	574
21.2	Liste d'arguments de champs . . . . .	594
21.3	Formats de sortie pour la date spécifiques à la langue . . . . .	598
21.5	Paramètres par défaut pour les termes dépendant des langues <i>english</i> et <i>ngerman</i> , tant que les paqs de sélection de langue n'ont pas été utilisés . . . . .	599



# Index

Commandes et Symboles
-----------------------

\\	114, 227, 230
\\*	114, 115, 227
\@@printindex	627
\@addtolengthlength	578
\@addtoplength	232, 577
\@bibbeginhook	570
\@bibendhook	570
\@currentt	379, 380, 382, 385, 426-428, 430, 432, 433, 450, 452, 507
\@currname	379, 380, 382, 385, 507
\@firstofone	433, 628
\@fontsizefilebase	550
\@hangfrom	565, 567
\@mainaux	417
\@mkboth	506, 513
\@mkdouble	506, 513
\@mkleft	506, 513
\@mkright	506, 513
\@newplength	232, 577
\@openbibcode	569
\@openbib@code	569, 570
\@partaux	417
\@printindex	627
\@ptsize	466
\@secntformat	509
\@secntmarkformat	509
\@sect	568
\@setlength	311, 551, 577, 582
\@setplength	577, 578, 580-589
\@ssect	568
\@startsection	567, 568
\@starttoc	418, 419, 427, 428, 431
\@tempskipa	562, 563
\@writefile	418, 419, 422
\color	551
\index	623
\police	
normalfont	252
\setlengthtoplength	157, 577
\usepackage	605
\@addpenalty	449
\@lowpenalty	449
\@openbib@code	569
\@setplength	585, 586
\@starttoc	450
\@tempb	348

- 
- `\@wrsindex` ..... 623  
`\@@wrsindex` ..... 623
- A**
- above** ..... 618  
**abstract** ..... 57, 105, 106  
`\abstract` ..... 57  
**Add**  
`\AddLayersAtBeginOfPageStyle` ..... 494, 495  
`\AddLayersAtEndOfPageStyle` 494  
`\AddLayersToPageStyleAfterLayer` ..... 492, 495  
`\AddLayersToPageStyleBeforeLayer` ..... 492  
`\AddLayersToPageStyle` 494  
`\AddToLayerPageStyleOptions` 496  
Addrfield ..... 195  
**add**  
`\addLevel` ..... 430  
`\addchap*` .. 57, 96, 97, 103, 138  
`\addchapmark` ..... 103, 105  
`\addchaptocentry` .... 549  
`\addchap` 57, 89, 96, 97, 103, 105, 553  
`\addcontentslinedefault` . 421  
`\addcontentslinetoeachtocfile` . 429, 430  
`\addcontentsline` .. 418, 422, 429, 430, 441, 548  
`\addmargin*` . 227, 228, 313, 314  
`\addmargin` ... 118, 227, 313  
`\addparagraphtocentry` . 549  
`\addpart*` ..... 97  
`\addparttocentry` .. 429, 549  
`\addpart` ... 96, 97, 105, 553  
`\addrchar` ..... 242, 243, 283  
`\addrentry` .... 240-243, 283, 284, 286  
`\addsec*` ... 96, 97, 103, 613  
`\addsecmark` ..... 103  
`\addsectiontocentry` ..... 549  
`\addsec` .... 82, 96, 103, 371, 553, 613  
`\addsubparagraphtocentry` ... 549  
`\addsubsectiontocentry` . 549  
`\addsubsubsectiontocentry` ... 549  
`\addtocentrydefault` 548, 549  
`\addtocontents` ..... 418, 421  
`\addtoeachtocfile` .. 428, 430  
`\addtokomafontgobblelist` .... 550, 551  
`\addtokomafontrelaxlist` 550, 551  
`\addtokomafont` ..... 44, 45, 52-54, 56, 69, 79-81, 85, 93, 108, 112, 113, 126, 127, 180, 195, 197, 199, 204, 206, 207, 224, 226, 227, 246, 250, 251, 254, 257, 267, 271, 295, 300, 301, 309-313, 358, 502, 525-527, 537, 539, 550, 551, 553, 557-559, 594, 595  
`\addtolength-valeur` ..... 24  
`\addtolengthplength` .... 157  
`\addtolength` ..... 156  
`\addtoeffields` ..... 589, 590  
`\addtotoclist` .. 424-427, 450, 468  
`\addvspace` ..... 442, 568

- 
- \addxcontentslinetoeachtocfile ..... 435
  - \addxcontentslinetoeachtocfile 429, 430
  - \addxcontentsline . 141, 423, 429, 430, 435, 441
  - \add ..... 553
  - \adrchar ..... 283
  - \asstocontents 423, 428, 429
  - add-on ..... 122, 149
  - addcontent= ..... 477
  - addEbene ..... 429
  - addheight= ..... 477
  - addhoffset= ..... 477
  - addmargin 118, 227, 228, 312, 313, 536
  - addmargin\* ... 118-120, 228, 312, 313
  - addniveau ..... 429
  - addrconv ..... 287, 288
  - address ..... 594
  - addressee . 176, 177, 195, 197
  - addresseeimage .... 152, 196, 198, 199
  - addresseimage ..... 198
  - addrfiel ..... 583
  - addrfiel=image ..... 583
  - addrfiel=PP ..... 583
  - addrfield .. 196, 198, 582, 583
  - addrfield=backgroundimage .. 199, 584
  - addrfield=image ..... 199
  - addrfield=PP . 197, 198, 584
  - addrfield=true 196, 197, 199
  - addtokomafont ..... 45, 176
  - addvoffset= ..... 477
  - addwidth= ..... 477
  - afteraddress ..... 183, 186
  - backaddress ... 152, 176, 185, 196, 197, 584
  - backaddress=valeur de .. 196
  - backaddressseparator ... 152, 196
  - backaddrheight 574, 583, 584
  - backgroundimage ..... 198
  - fromaddress .... 176, 183, 186
  - PPBackGroundImage ... 198
  - PPBackgroundImage .... 198
  - PPbackgroundimage .... 198
  - ppBackGroundImage ... 198
  - ppBackgroundImage .... 198
  - ppbackgroundimage ..... 198
  - scraddr ..... 283, 286, 287
  - specialmail ..... 197
  - toaddress ..... 195-197
  - toaddrheight .. 575, 577, 582, 595
  - toaddrhpos .... 235, 576, 577, 582, 595
  - toaddrindent .. 199, 577, 583
  - toaddrvpos ... 576, 577, 582, 595
  - toaddrwidth ... 235, 577, 582, 595
  - toadress ..... 196
  - adr**
  - , ppExplicite ..... 198
  - \adrchar ..... 242, 243, 283
  - \adrentry 239-243, 283, 284, 286
  - adrconv ... 240, 242, 243, 288
  - adrdiir.tex ..... 243
  - adrFreeIVempty ..... 286
  - adrFreeIVshow ..... 286
  - adrFreeIVstop ..... 286
  - adrFreeIVwarn ..... 286
  - fromzipcode ..... 198

- PP ... 153, 154, 195, 197, 198
- pp ..... 198
- PPcode ..... 154, 198
- PPdatamatrix ..... 153, 198
- PPExplicite ..... 198
- PPexplicite ..... 198
- specialmail ..... 154
- After**
- \AfterAtEndOfClass ... 405, 406
- \AfterAtEndOfPackage 405, 406
- \AfterBibliographyPreamble . 148
- \AfterClass! ..... 405
- \AfterClass! ..... 406
- \AfterClass\* ..... 405
- \AfterClass+ ..... 405
- \AfterClass ..... 405
- \AfterFile ..... 405, 407, 408
- \AfterPackage! ..... 405
- \AfterPackage! ..... 406
- \AfterPackage\* ..... 405
- \AfterPackage+ ..... 405
- \AfterPackage ..... 405
- \AfterReadingMainAux 408, 410
- \AfterSelectAnyPageStyle .... 487
- \AfterStartingTOC 421, 432, 433, 451
- \AfterTOCHead .... 19, 432
- AfterSettingArea ..... 546
- alias** ... xviii, 253, 487-489, 491, 492, 495-497, 550, 591
- \ProvidePageStyleAlias . 488
- \RedeclarePageStyleAlias .... 488
- alinéa** . 57, 63, 64, 99, 100, 262, 317, 319-327, 331, 336, 339, 342-344, 347, 353, 359, 361, 364, 365, 374, 443, 473, 534, 548
- allintoone** ..... 606, 607
- annexe** ..... 93
- appendix** ..... 144
- \appendixmore .... 568, 569
- \appendixname ..... 400
- \appendixprefix ..... 93
- \appendix ..... 88, 144, 568
- appendixprefix ... 84, 85, 87, 144, 568, 569
- appendixprefix=true . 46, 85
- appendixwithoutprefix ... 86
- appendixwithoutprefixline 86
- appendixwithprefix ..... 87
- appendixwithprefixline ... 87
- noappendixprefix ..... 86
- onlineappendix ..... 86
- area**
- \AfterCalculatingTypearea\* .. 546
- \AfterCalculatingTypearea ... 530, 546
- \AfterRestoreareas\* .... 545
- \AfterRestoreareas ..... 545
- \AfterSettingArea\* ..... 546
- \AfterSettingArea ..... 546
- \BeforeRestoreareas\*\* .. 545
- \BeforeRestoreareas .... 545
- \activateareas ..... 544-546
- \areaset ..... 27, 28, 34, 544
- \recalctyparea ..... 23
- \recalctypeaarea ..... 19
- \recalctypearea .. 20, 21, 24, 26, 27, 30, 43, 174, 294
- \storeareas ..... 545

- 
- \typearea ..... 19
  - area= ..... 478
  - areaset ..... 18, 544, 546
  - areasetadvanced ..... 544
  - heigh ..... 478
  - hoffset ..... 478
  - storeareas ..... 546
  - textarea ..... 482
  - typearea ..... 544
  - typearea ..... 2, 7-9, 11-14, 16, 17, 20, 21, 23-30, 33, 34, 39, 41, 63, 67, 158, 161, 176, 216, 236, 248, 272, 381, 394, 485, 498, 543-546, 574, 592
  - typearea.cfg ..... 546
  - typearea.sty ..... 2, 522, 543
  - voffset ..... 478
  - width ..... 478
  - arguments**
  - \Address ..... 284
  - \Comment ..... 284
  - \FirstName ..... 284
  - \FreeIII ..... 284
  - \FreeII ..... 284
  - \FreeIV ..... 284, 286
  - \FreeI ..... 284
  - \LastName ..... 284
  - \Name ..... 284
  - \Telephone ..... 284
  - artcl** ..... 289
  - At**
  - \AtAddToTocList .. 426, 435
  - \AtBeginDocument 165, 397, 399, 404, 487
  - \AtBeginLetter .... 165, 166
  - \AtEndBibliography .... 148
  - \AtEndDocument .. 409, 410
  - \AtEndLetter ..... 165
  - \AtEndOfClass .... 165, 406
  - \AtEndOfFamilyOptions 386
  - \AtEndOfPackage ..... 406
  - attribut**
  - addContents ..... 483
  - addcontents ..... 485
  - contents ..... 483, 485
  - pretocontents ..... 483, 485
  - autorisation** ..... xviii, 233
- B
- babel** xv, 51, 118, 144, 162, 200, 235, 279, 299, 324, 364, 369, 370, 398, 423, 426, 428, 434, 449, 450, 466, 506, 513, 529, 597, 599
  - BCOR** ..... 7, 8, 19, 20, 27, 272
  - BCOR=0 pt ..... 8
  - BCOR=correction ..... 12
  - Before**
  - \At@startsection .. 567, 568
  - \BeforeClass ..... 405, 411
  - \BeforeClosingMainAux ..... 408-410, 487
  - \BeforeFamilyProcessOptions 382, 383
  - \BeforeFile ..... 405
  - \BeforePackage .... 405, 411
  - \BeforeSelectAnyPageStyle ... 487
  - \BeforeStartingTOC ... 432, 433, 451
  - \BeforeTOCHead ..... 432
  - \Before@sect ..... 567, 568
  - \Before@ssect ..... 567, 568
  - \Before@@sect ..... 568
  - below** ..... 186
  - bibliography** ..... 59, 145, 146

`\newbibstyle` ..... 145  
`\AfterBibliographyPreamble` .  
     570  
`\AtEndBibliography` .... 570  
`\BreakBibliography` 147, 570  
`\bibliography`  
     `newblock` ..... 570  
`\bib@beginhook` ... 569, 570  
`\bib@endhook` ..... 569, 570  
`\newbibstyle` .. 146, 569, 570  
`\newblock` 145, 147, 569, 570  
`\setbibpreamble` ..... 146  
`bib` ..... 59  
`bibliography=openstyle` . 570  
`bibliographynumbered` ... 58,  
     59, 145  
`leveldown` ..... 145  
`natbib` ..... 146  
`nobib` ..... 60  
`nobibliography` ..... 60, 145  
`noidx` ..... 60  
`noindex` ..... 60  
`nototoc` ..... 145  
`numberedbibliography` ... 59  
`oldstyle` ..... 145  
`openbib` ..... 569  
`openstyle` ..... 145, 147  
`réglage` ..... 144  
`thebibliography` .... 146, 147  
`totoc` ..... 145, 146  
`totocnumbered` ..... 145, 146  
**book** ... 37, 84, 92, 96, 262, 289,  
     297, 437, 438, 440, 457, 472,  
     473, 511, 547, 610  
**botcenter** ..... 216  
**both** ..... 319  
**box**  
     `\colorbox` ..... 522  
     `\dp` ..... 522

`\fps@remarkbox` ..... 456  
`\layerheight` ..... 522  
`\layerwidth` ..... 522  
`\listofremarkboxes` ..... 456  
`\listremarkboxnames` ... 456  
`\remarkbox*` ..... 457  
`\remarkboxformat` ..... 456  
`\remarkboxname` ..... 456  
`\remarkbox` ..... 457  
`\strutbox` ..... 522  
`\theremarkbox` 453, 456, 457  
`colorbox` ..... 567  
`remarkbox` 453, 454, 456, 457  
`remarkbox*` ..... 456  
**boxed** ..... 133

C

**calc** ..... 16  
**caption** ..... 45, 126, 130  
     `\captionSprache` ..... 397  
     `\captionaboveof` .... 127, 128  
     `\captionabove` 122, 125, 129,  
         133, 455  
     `\captionbelowof` ..... 127  
     `\captionbelow` 122, 125, 129,  
         133, 455  
     `\captionbeside` ..... 455  
     `\captionformat` .... 133, 134  
     `\captionof` ..... 127, 133  
     `\captionsUKenglish` .... 598  
     `\captionsUSenglish` 398, 598  
     `\captionsacadian` ..... 598  
     `\captionsamerican` ..... 598  
     `\captionsaustralien` .... 598  
     `\captionsaustrian` ..... 598  
     `\captionsbritish` ..... 598  
     `\captionscanadian` ..... 598  
     `\captionscanadien` ..... 598

- 
- `\captionscroatian` ..... 598
  - `\captionsczech` ..... 598
  - `\captionsdutch` ..... 598
  - `\captionsenglish` ..... 598
  - `\captionsfinnish` ..... 598
  - `\captionsfrançais` ..... 598
  - `\captionsfrench` ..... 598
  - `\captionsgerman` ..... 598
  - `\captionssitalian` ..... 598
  - `\captionsslanguange` ..... 399
  - `\captionssnaustrian` ..... 598
  - `\captionssnewzealand` .... 598
  - `\captionssngerman` .. 398, 598
  - `\captionssnorsk` ..... 598
  - `\captionssnswissgerman` .. 598
  - `\captionsspolish` ..... 598
  - `\captionsslovak` ..... 598
  - `\captionssspanish` ..... 598
  - `\captionssswedish` ..... 598
  - `\captionssswissgerman` ... 598
  - `\captions` ..... 397
  - `\caption` .. 122, 125, 127-129, 133, 407, 453, 455, 456
  - `\defcaptionname*` ..... 398
  - `\defcaptionname` ..... 398
  - `\newcaptionname*` ..... 398
  - `\newcaptionname` ..... 398
  - `\providecaptionname*` .. 398
  - `\providecaptionname` ... 369, 398
  - `\renewcaptionname*` .... 398
  - `\renewcaptionname` ..... 398
  - `\setcapmargin*` .... 136, 137
  - `\setcapmargin` ..... 136, 137
  - `\setcapwidth` ..... 136
  - `capt-of` ..... 127
  - `caption2` ..... 407
  - `caption=nooneline` ..... 137
  - `caption=reglage` ..... 122
  - `captionabove` ..... 127
  - `captionbelow` ..... 127
  - `captionbeside` . 123-125, 131, 133
  - `captionlabel` ..... 45, 126
  - `captionofbeside` ..... 133
  - `captions` ..... 123, 130
  - `captions=figureheading` . 124
  - `captions=figuresignature` ..... 122, 124
  - `captions=heading` .. 122, 123
  - `captions=leftbeside` ..... 124
  - `captions=nooneline` ..... 123
  - `captions=oneline` ..... 123
  - `captions=rightbeside` .... 124
  - `captions=signature` 122, 123, 125
  - `captions=tableheading` .. 124
  - `captions=topbeside` ..... 131
  - captions**
    - `captions=figureheading` . 122
  - cc** ..... 599
  - ccseparatot** ..... 171
  - CenturyPart** ..... 275
    - `\CenturyPart` ..... 275
  - chapitre** ..... 548
  - chapter** .. 45, 46, 62, 91, 93, 97, 99, 103, 144, 249, 250, 264, 270, 421, 437, 439, 440, 442, 443, 500, 553, 558, 560, 566
    - `\addchapter*` ..... 566
    - `\addchap` ..... 566
    - `\chapappifchapterprefix` 102, 104
    - `\chapapp` ..... 102, 104
    - `\chapter*` ... 57, 96, 97, 467, 566, 610
    - `\chaptermarkformat` .... 103, 105, 267, 502

<code>\chaptermark</code> .	103, 268, 503, 505
<code>\chapternumdepth</code> .....	104
<code>\chapterpagestyle</code> .....	97
<code>\chapterprefix</code> .....	93
<code>\chapter</code> .....	88, 90, 92, 93, 96, 101, 105, 436, 472, 504, 509, 536, 553, 555, 560, 562, 566, 610
<code>\if@chapter</code> .....	472
<code>\section</code> .....	96
<code>chapteratlists</code> .....	89
<code>chapterentry</code> .	45, 63, 139, 140
<code>chapterentrydots</code> ..	46, 61, 63
<code>chapterentrypagenumber</code>	46, 62, 63
<code>chaptergapline</code> .....	140
<code>chaptergapsmall</code> .....	140
<code>chaptermark</code> .....	103, 512
<code>chapterprefix</code> .	46, 84, 85, 88, 102, 144, 560, 563
<code>chapterprefix=true</code> ..	46, 102
<code>chapterwithprefix</code> .....	88
<code>chapterwithprefixline</code> .....	88
<code>entryprefix</code> .....	140
<code>flat</code> .....	140
<code>graduated</code> .....	140
<code>ignorechapter</code> .....	141
<code>indent</code> .....	140
<code>indented</code> .....	140
<code>indenttextentries</code> .....	141
<code>left</code> .....	140
<code>leftaligntextentries</code> .....	141
<code>leveldown</code> .....	141
<code>nochaptergap</code> .....	141
<code>onelinechapter</code> .....	85
<code>onelinechaptergap</code> .....	140
<code>paragraphe</code> .....	504
<code>raggedchapter</code> .....	565
<code>smallchaptergap</code> .....	140
<code>twolinechapter</code> .....	85, 88
<code>withchapterentry</code> .....	140
<b>checkinstall</b> .....	324
<b>classe</b>	
<code>\LoadClassWithOptions</code>	411
<code>\LoadClass</code> .....	405, 411
<b>Clause</b> .	317, 320, 321, 325, 326, 357, 358, 360, 366, 373
<code>\Clause</code> .....	366
<code>\SubClause</code> .....	366
<code>Clause=instruction</code> .....	366
<code>Clauseformat</code> .....	359
<code>ClauseNumberFormat</code> ..	326, 368
<code>contract.Clause</code> .....	359
<code>SubClause</code>	321, 326, 357, 358, 360, 374
<code>SubClause=instruction</code> ..	366
<b>clause</b> ..	320, 358, 359, 366, 373
<code>ClauseOnly</code> .....	367
<code>clauseonly</code> .....	367
<code>OnlyClause</code> .....	367
<code>onlyclause</code> .....	367
<b>cleardoublepage</b> .....	220
<b>code</b>	
<code>fromzipcode</code> .....	197
<code>zipcodeseparator</code> .....	197
<b>color</b> .....	296
<b>column</b> .....	143
<b>command</b>	
<code>\DeclareSectionCommand</code> ....	563
<code>\RedeclareSectionCommand</code> .	563
<code>\newcommand*</code> ...	359, 415, 511
<code>\newcommand</code>	101, 454, 508, 546, 570



- 
- `\renewcommand*` ..... 215
  - `\renewcommand` . 49, 71, 99-101, 104, 107, 108, 110, 134, 225, 298, 311, 312, 318, 363, 568, 570, 595
  - commande**
    - `\InputAdressFile` ..... 283
    - `\autodot` . 100, 101, 267, 502
    - `\backmatter` 82, 83, 472, 473
    - `\begin document` ..... xx-vii, 421, 473, 545, 599
    - `\begin environnement` ... xx-vii
    - `\begin letter` ..... 591
    - `\begin` ..... xx-vii, 611, 625
    - `\cc` ..... 171
    - `\cehead` ..... 259
    - `\chead` ..... 259
    - `\cohead` ..... 259
    - `\extendtheindex` ..... 611
    - `\frontmatter` ..... 472, 473
    - `\g@addto@macro` ..... 401
    - `\hspace*` ..... 567
    - `\if@mainmatter` ... 472, 473
    - `\ihead` ..... 259
    - `\lehead` ..... 259
    - `\lohead` ..... 259
    - `\lowercase` ..... 628
    - `\l@addto@macro` ..... 401
    - `\mainmatter` 82, 83, 472, 473
    - `\ohead` ..... 259
    - `\opening` . 162, 164, 167, 216, 217, 234, 286, 581, 585, 586, 588, 593, 597
    - `\protect` ..... 532
    - `\rehead` ..... 259
    - `\relax` ..... 508
    - `\renewcommand` ..... 311
    - `\rohead` ..... 259
    - `\secddef` ..... 567
    - after begin ..... 611
    - after end ..... 611
    - autoremoveinterface ..... 508
    - autoremoveinterfaces .... 507
    - before begin ..... 611
    - before end ..... 611
    - catcodes ..... 628
    - forceoverwrite ..... 508
    - forceoverwrite=on-off-valeur .. 507
    - idxcommand ..... 622
    - includegraphics ..... 199
  - compteur**
    - `\newmaxpar` ..... 327
    - thefigure ..... 134
    - thetable ..... 134
  - contract** 317, 318, 321, 325, 326, 328, 334-336, 343, 357-359
    - `\contract@paragraph` ... 335
    - `\contract` ..... 334
    - `\thecontractClause` ..... 326
    - `\thecontractSubClause` .. 326
    - contract.Clause .... 358, 359
    - contractAbsoluteParagraph .. 334
    - contractParagraph ..... 334
    - contractSubParagraph .. 334
    - contract@everypar ..... 343
    - contract@Paragraphformat .. 334
    - Paragraphformat ..... 334
    - thecontractParagraph ... 334
    - thecontractSubParagraph .... 334
  - coplain** ..... 71
  - counter**
    - `\refstepcounter` ..... 555

`\stepcounter` ..... 555  
`counterwithin` ..... 555  
**co@ptsize** ..... 466  
**codictum** ..... 46  
**copolice**  
`texttt` ..... xx-  
vii  
**cosection** ..... 46, 319  
**cousepackage** ... 597, 604, 606,  
607  
**CTAN** ..... xxix, 316, 636  
**current** ..... 303

**D**

**date** ..... 46  
**Declare**  
`\DeclareLayer` ..... 476  
`\DeclareNewAliasPageStyle` ..  
489  
`\DeclareNewJuraEnvironment`  
325, 366  
`\DeclareNewLayer` . 476, 486  
`\DeclareNewNoteColumn` ....  
527, 537, 539  
`\DeclareNewPageStyleAlias` ..  
488  
`\DeclareNewPageStyleBy-`  
Layers ..... 485, 489  
`\DeclareNewSectionCom-`  
mands ..... 562  
`\DeclareNewSectionCommand`  
58, 552, 553, 561  
`\DeclareNewTOC` . 448, 455,  
457-459  
`\DeclareNoteColumn` ... 527,  
537-539  
`\DeclareOption` ..... 382-384  
`\DeclarePageStyleAlias` . 488

`\DeclarePageStyleByLayers` ..  
489, 521  
`\DeclareRobustCommand*` ...  
511  
`\DeclareRobustCommand` ....  
400  
`\DeclareSectionCommandsau-`  
ter ..... 562  
`\DeclareSectionCommand` ....  
58, 440, 552-554, 561, 562,  
568  
`\DeclareSectionNumberDepth`  
473  
`\DeclareTOCEntryStyle` 440,  
441, 446, 448  
`\DeclareTOCStyleEntry` 438,  
440, 441, 446  
`\RedeclareNoteColumn` 527,  
537, 539  
`DeclareTOCStyleEntry` . 437  
**dedication** ..... 46  
**deffootnote** ..... 80  
**DefineTOC**  
`\DefineTOCEntryBooleanOp-`  
tion ..... 446  
`\DefineTOCEntryComman-`  
dOption ..... 446  
`\DefineTOCEntryIfOption` ...  
446  
`\DefineTOCEntryLengthOp-`  
tion ..... 446  
`\DefineTOCEntryNumberOp-`  
tion ..... 446  
`\DefineTOCEntryOption` .....  
446, 447  
**descriptionlabel** ..... 45, 46  
**dictum** ..... 46, 312  
**dictumauthor** ..... 46, 312  
**dictumtext** ..... 46

- 
- disposition** ..... 46
- DIV** .. 7-9, 13-19, 23, 24, 26, 27, 30, 34, 485
- `\DIV=calc` ..... 15
  - `\DIV=facteur` ..... 13, 15
  - `DIV` ..... 16
  - `DIV=areaset` ..... 28
  - `DIV=calc` ..... 14-16, 21, 34
  - `DIV=classic` ..... 14-16
  - `DIV=classique` ..... 484
  - `DIV=current` ..... 16
  - `DIV=facteur` ..... 13, 15, 16
  - `DIV=last` ..... 16, 17, 28
- documentclass** ..... 246
- `\documentclass` ... 9, 10, 16, 17, 21, 37-39, 75, 121, 137, 158, 160, 178, 233, 234, 282, 290, 291, 297, 305, 357, 378, 382, 405, 411, 414, 464, 592
  - `documentclass` 247, 464, 470, 471
- dummy** . 320, 336, 359, 360, 373, 447
- `ifcontract@dummy` ..... 336
- DVI** ..... xxxi, 31, 32, 394
- dvips** ..... 407, 408
- E**
- ellipsispar** .. 321, 333, 362, 363
- empage** xvi, 2, 4, 22, 27, 34, 41, 42, 143, 167, 314, 498, 552, 576, 585, 586, 589, 590
- empagement** ..... xvi, 4, 5, 23
- encap** ..... 623
- entête** . 47, 48, 51, 60, 152, 153, 245, 581, 613
- environnement**
- `Sentence=instruction` ... 366
  - `Sentence` ..... 366
- eso-pic** ..... 484, 593
- expéditeur** .. 152-155, 167, 176, 177, 183-189, 191-194, 198, 200, 201, 203, 207, 210, 212, 216, 218, 232, 235, 238, 239, 550, 575, 576, 581-585, 594
- extendedfeature=possibilité** .. 293
- extendedfeature=title** .. 296, 297, 303
- F**
- false** 20, 23, 24, 31, 32, 41, 49, 57, 65, 85, 102, 161, 179, 182, 183, 185-187, 192, 198, 199, 201, 209, 293, 297, 318, 319, 362, 381, 386, 466-468, 588, 590
- famille**
- `boucher` ..... 380
  - `boudin.sty` ..... 380
  - `merguez.sty` ..... 380
  - `salami.sty` ..... 380
- Family**
- `\FamilySetBool` ..... 388
  - `\FamilySetNumerical` ... 389
  - `\FamilyUnknowKeyValue` .... 389
- FamilyKey**
- `\DefineFamilyKey` ..... 389
  - `\FamilyKeyStateProcessed` ... 389, 392
  - `\FamilyKeyStateUnknowValue` ..... 391
  - `\FamilyKeyStateUnknowKeyValue` ..... 389

- 
- `\FamilyKeyState` .. 389, 392, 393
  - `\FamilyNumericalKey` .. 389
  - `FamilyKeyStateUnknownValue` 390
  - fancyhdr** ..... 246, 411
  - fichier**
    - `.cls` ..... xxxv, xxxvi, 406
    - `.doc` ..... xxxv
    - `.html` ..... xxxvi
    - `.ico` ..... xxxv, xxxvi, 198
    - `.pdf` ..... xxxvi
    - `.snlc` ..... 532
    - `.source` ..... xxxv
    - `.sty` ..... xxxv, xxxvi, 406
    - `.tex` ..... xxxv
    - `.txt` ..... xxxvi
    - `\InputIfFileExists` .. 405, 410
    - `\SelectinputMappings` .... 99
    - `\ifstr` ..... 388
    - `\inputAddressFile` ..... 285
    - `\input` ..... 240, 405, 539
    - `\read` ..... 539
    - `\selinput` ..... 530
    - `fig.` ..... 398
    - `input` ..... 99
    - `inputenc` ..... 235, 370
    - `readme` ..... xxxvi
    - `scrmain.ins` ..... xxxv
    - `stoc` ..... 421
    - `texconfig` ..... xxxv
    - `texhash` ..... xxxv
  - figure** ..... 6, 29, 38, 58, 66, 76, 82, 121, 122, 125-132, 134, 135, 138-142, 162, 190, 193, 209, 212, 229, 231, 235, 324, 369, 398, 400, 417, 423, 426, 435, 436, 438-440, 443, 444, 452-460, 465, 475, 518
  - `\figureformat` ..... 456
  - `\figurename` ..... 456
  - `\fps@figure` ..... 456
  - `\ifpsoutput` ..... 394
  - `\list@figurename` ..... 456
  - `\scr@ifpsoutput` ..... 394
  - `\thefigure` ..... 456
  - `abovefigure` ..... 123
  - `belowfigure` ..... 123
  - `bottomatfigure` ..... 123
  - `figure*` ..... 456, 458
  - `figureabove` ..... 123
  - `figurebelow` ..... 123
  - `figureformat` ..... 134
  - `figureheading` ..... 122, 123
  - `figuresignature` ..... 122-124
  - `topatfigure` ..... 123
  - float** ..... 434, 456, 465
    - `\float@addtolists` ..... 465
    - `\float@listhead` ..... 465
    - `floattype` ..... 456
  - foo** . 379, 405, 415, 424-427, 429, 608, 618
  - footnote** ..... 46
  - footnotelabel** ..... 46
  - format** ..... xv, xxviii, xxxiii, 1, 3, 7, 12-15, 22, 29-32, 38, 41, 42, 55, 59, 69, 85, 101, 111, 134, 145, 147, 161, 162, 225, 236-239, 246, 282, 287, 300, 331, 365, 367, 421, 442, 443, 445, 467, 544, 545, 554, 563, 566, 578, 580, 583, 592, 593, 595, 598, 606, 610
    - `\chapterformat` ..... 100
    - `\othersectionlevelsformat` ..... 100
    - `\parformatseparation` ... 361
    - `\parformat` ..... 361

- 
- `\partformat` ..... 100
  - `\sectionformat` ..... 100
  - `\subparagraphformat` ... 100
  - `\subsectionformat` ..... 100
  - `\subsubsectionformat` ... 100
  - `nomformat` ..... 554
  - `bib` ..... 287
  - `chapterformat` .. 87, 88, 100, 101
  - `chapterlinesformat` ..... 102, 563-565
  - `chapterlineswithprefixformat` . 102, 563-565
  - `chaptermarkformat` . 246, 504
  - `compact` ..... 145
  - `dictumauthorformat` 108, 312
  - `entryformat` ..... 440
  - `entrynumberformat` ..... 440
  - `othersectionlevelsformat` 100
  - `parciteformat` ..... 352
  - `parcite@format` ..... 331
  - `parformat` ..... 361, 362
  - `parformatseparation` ... 362
  - `partformat` ..... 104
  - `remarkboxformat` ..... 457
  - `sectionformat` ..... 101
  - `sentenceciteformat` . 327, 352
  - `sentencecite@format` ... 331
  - `tableformat` ..... 134
  - `textchaptermarkformat` . 103
- G**
- Gnu-make** ..... xxxiv
  - graphics** ..... 199
  - graphicx** ..... 199
- H**
- hyperindex** ..... 625
  - hyperpage** ..... 623
  - hyperref** .... 318, 323, 338, 346, 347, 350, 356, 399, 406-408, 417, 434, 467, 623, 625
    - `hyperref-option` ..... 625
    - `hyperref.cfg` ..... 408
    - `hyperref=false` ..... 467
  - hyperref\*** ..... 347
  - hyperref.cfg** ..... 407
  - hypertex** ..... 407, 408
- I**
- Ico** ..... 236, 237
  - Ico-scrletter** ..... 592
  - if**
    - `\IfUsePrefixLine` ... 100, 101
    - `\ifcontract@dummy` ... 336
    - `\ifkomavareempty*` .. 156, 590
    - `\ifkomavareempty` ... 156, 590
    - `\ifkomavar` ..... 155
    - `\ifnumbered` ..... 103
    - `\if@juratitlepagebreak` .. 329
    - `ifiscounter` ..... 396
  - image**
    - `backgroundimage` ..... 195
    - `toaddress` ..... 195
    - `toname` ..... 195
  - indent** ... 60, 438, 440, 443, 458, 459, 553, 556-558, 561
  - indented** ..... 60
  - indenttextentries** ..... 60
  - index** 60, 607, 609, 614, 620-622, 629
    - `.idx` ..... 605, 606
    - `.ind` ..... 605
    - `\AtNextWriteToIndex` .. 607
    - `\AtWriteToIndex` ..... 607

<code>\DeclareNewPageStyleBy-</code>	
<code>Layers</code> .....	492
<code>\DeclarePageStyleByLayersin-</code>	
<code>dex</code> .....	492
<code>\DeclarePageStyleByLayers</code> ..	492
<code>\indexentry</code> ...	602, 615, 619
<code>\indexname</code> .....	610
<code>\indexshortcut</code> .....	611
<code>\index@preamble</code> .....	610
<code>\index@raccourci@nom</code> .	627
<code>\index</code> .....	602, 606, 607
<code>\jobname.idx</code> .....	602
<code>\makeindex</code> .....	417, 605
<code>\newindex</code> ....	602, 605, 606, 609, 622, 629-631
<code>\newprotectedindex</code>	608, 630
<code>\printindex*</code> .....	612, 626
<code>\printindex</code> .....	602, 605, 609-611, 613, 618, 626, 628, 631-633
<code>\printsindex*</code> ....	613, 632
<code>\printsindex</code>	605, 610, 611, 632
<code>\setindexpreamble</code> .....	631
<code>\sindex</code> ...	602, 605-608, 621, 623
<code>\useindexpreamble</code> .....	611
<code>\useindexpreamblr</code> .....	610
<code>\usepackage</code> .....	605, 606
<code>alias</code> .....	606
<code>allabout-ani.ind</code> .....	618
<code>allabout-fru.ind</code> .....	618
<code>allabout-idx.ind</code> .....	618
<code>allabout-veg.ind</code> .....	618
<code>allintoone</code> .....	606
<code>idx</code> .....	60, 605, 613
<code>index</code> .....	606
<code>index=default</code> .....	148
<code>index=totoc</code> .....	148
<code>indexentry</code> .....	607
<code>indexnumbered</code> .....	60, 149
<code>indexpagestyle</code> .....	149
<code>indexshortcut</code> .....	627
<code>index@preamble</code> .....	627
<code>index@raccourci@preamble</code> ..	631
<code>kvtcb@text@index</code> .	610, 627
<code>leftaligntextentries</code> .....	60
<code>leftalignunnumbered</code> .....	60
<code>makeidx</code> .....	601, 602
<code>MakeIndex</code> .....	614
<code>makeIndex</code> .....	602
<code>makeindex</code> .....	605
<code>name</code> .....	605
<code>options</code> .....	621
<code>printindex</code> .....	626
<code>ProvidePageStyleByLayers</code> ...	492
<code>split</code> .....	603, 605
<code>splitidx</code> ...	148, 601-607, 610, 611, 618, 620, 621
<code>splitidx.dtx</code> .....	601
<code>Splitindex</code> .....	603
<code>splitindex</code> .	604, 606, 614, 618
<code>splitindex.java</code> .....	619
<code>splitindex.c</code> .....	604
<code>splitindex.class</code> .....	619
<code>splitindex.jar</code> .....	619
<code>splitindex.java</code> .....	603
<code>splitindex.pl</code> ...	603, 616, 619
<code>splitindex.tex</code> .	604, 615, 619, 620
<code>splitindex.tlu</code> ..	604, 615, 616
<code>splitindx</code> .....	608
<code>splitted</code> .....	614, 623, 631
<code>tcolorbox</code> .....	610

theindex . 610, 611, 613, 626,  
631  
useindex ..... 606, 607, 621  
**installD** ..... xxxiii

## J

**jura**  
juratitlepagebreak . 318, 329,  
360  
jura@everpar ..... 343  
jura@everypar ..... 342

## K

**KDP** ..... xxxi  
**key**  
  \DefineFamilyKey ..... 380  
  \DefineFamilykey ..... 385  
  \FamilyKeyStateNeedValue ..  
  380  
  \FamilyKeyStateProcessed ...  
  380  
  \FamilyKeyStateUnknownVa-  
  lue ..... 380  
  \FamilyKeyStateUnknown ...  
  380  
  \FamilyKeyState ..... 380  
  \define@key ..... 380  
  \setkeys ..... 321, 336  
  key=value ..... 378, 537  
  key=value-liste ..... 320  
  keyval 236, 335, 378-380, 382  
  prioritykey ..... 197  
**keyval**  
  xkeyval ..... 378  
**kile** ..... 324  
**KOMA**  
  \KOMAOptions=DIV ... 19

\KOMAOptions ..... 10,  
12-14, 16, 17, 19, 21, 23-25,  
27, 28, 30, 39, 41, 76, 121,  
131, 137, 158, 160, 164, 222,  
247, 282, 291, 292, 304, 305,  
307, 318, 323, 357, 358, 455,  
465, 468  
\KOMAoption .... 10, 12-14,  
16, 19, 21, 23-25, 27, 30, 39,  
76, 121, 137, 158, 222, 282,  
291, 292, 307, 323, 357, 358,  
465, 468, 471  
komaabove ..... 123, 133  
komabelow ..... 133  
KOMAold 230, 237, 581, 588,  
600  
KOMAold.ico ..... 232  
KOMAoption .... 39, 41, 160,  
247, 465, 471  
KOMAOptions .. 39, 247, 465,  
471  
KOMAOptions=BCOR .. 19

## L

**labeling** ..... 312  
**labelinglabel** ..... 46  
**labelingseparator** ..... 47  
**langue** .....  
iii, xv, xvii, xviii, xxv, xx-  
viii, xxxi, 6, 33, 56, 74, 84,  
93, 118, 121, 140, 162, 189,  
200, 235, 278, 279, 315, 316,  
326, 369, 375, 397-400, 420,  
423, 434, 449, 450, 455, 466,  
475, 506, 513, 529, 569, 597-  
599, 636  
\extralanguage ..... 399  
\selectlanguage ..... 597

- 
- american ..... 369
  - anglais
    - m.eik ..... 353
  - austrian ..... 369
  - british ..... 369
  - canadian ..... 369
  - english ..... 369, 599
  - français ..... xx-  
vii, xxxv, 84, 200, 204, 265,  
279, 320, 370, 397, 523, 597,  
599, 636
  - french xv, 200, 279, 370, 398,  
597
  - french.ldf ..... 404
  - frenchb ..... 200, 398
  - naustrian ..... 369
  - ngerman ..... 369, 599
  - UKenglish ..... 369
  - USenglish ..... 369, 398
  - Latex2e** ..... 317, 403
  - legal.txt** ..... 622
  - length**
    - \@setlength ..... 107, 232
    - \newlength ..... 156
    - \setlength ..... 156
    - \thelength ..... 156
    - \unitfactor ..... 596
    - \unitlength ... 484, 595, 596
    - \useplength ..... 157
  - level** ..... 440, 443, 553, 556-558,  
560-562
    - increaselevel ..... 562
    - increaselevel=entier ..... 562
    - level=niveau hiérarchique ....  
459
    - tolevel ..... 459, 460, 562
    - tolevel=valeur ..... 459
  - linefill** .. 440, 442, 443, 445, 449
  - listing** ..... xx-  
vii
  - listname** ..... 456
  - listof** ... 58, 60, 88, 140, 142, 436
    - \box
      - listofremarkboxes ..... 454
    - \listoeffigures ..... 605
    - \listoftables ..... 605
    - \listofaechtoc ..... 432
    - \listofeachtoc . 431, 432, 434,  
451
    - \listoffigures .. 142, 417, 456
    - \listofflofentryname ..... 139
    - \listofflotentryname ..... 139
    - \listofremarkboxes ..... 457
    - \listofstoc ..... 421
    - \listoftables ..... 142
    - \listoftoc\* 431, 432, 450, 451
    - \listoftoc ..... 431, 432, 434,  
450-452, 454
    - \listof ..... 420, 460
    - default ..... 149
    - graduated ..... 138
    - leveldown . 145, 147, 420, 434,  
437
    - listof=chapterentry ..... 139
    - listof=chaptergapline ... 139
    - listof=chaptergapsmall .. 89,  
139
    - listof=entryprefix ..... 139
    - listof=flat ..... 139
    - listof=leveldown ... 138, 434
    - listof=nochaptergap .... 139
    - listof=notoc ..... 138
    - listof=numbered ... 138, 435
    - listof=réglage ..... 138
    - listof=totoc ..... 138, 436
    - listoffigures ..... 418
    - listofgestionation ..... 420



- 
- listofnumbered ..... 58, 142
  - listoftables ..... 417
  - nolistof ..... 61
  - nonumberline ..... 435
  - nottotoc ..... 149
  - numberline ..... 435
  - oldstyle ..... 147
  - plainheading ..... 149
  - stoc ..... 421
  - toc ..... 149
  - totoc ..... 149
  - lof** ..... 139
  - longtable** ... 122, 137, 403, 404
    - origlongtable ..... 137
  - lor** ..... 454, 457
- M**
- MacTeX** ..... xxix
  - make** ..... 324
    - \MakeMarkcase 70, 265, 266, 270, 433, 500, 501
    - \MakeUppercase .. 266, 433, 434, 442, 501, 510, 551, 565, 590
    - make install ..... xxxv
    - make install ..... 324
  - makeatletter** ..... 567
  - makeatother** ..... 567
  - makeidx** 604, 608, 609, 622, 627
  - makeindex** . 607, 615, 616, 618, 622
  - maketitle** ..... 54, 57
  - MakeUppercase** ..... 266, 501
  - manual** ..... 199
  - marge** ..... xvi, xxvii, 3-5, 7, 12, 13, 16, 19-25, 28, 30, 33, 34, 41, 49, 55, 102, 119, 137, 142, 143, 161, 173, 174, 178, 213, 227-229, 249, 250, 265, 270, 272, 273, 294, 298, 301, 313, 314, 435, 499-501, 503, 523, 530, 535, 537, 544-546, 552, 554, 557, 566, 567, 580, 583, 588, 594
  - .ico ... 179, 200, 230, 232-236, 578, 580-582, 584-589, 592, 593
    - DIN.ico ..... 232
    - DINmtext ..... 236, 578, 582
    - DINmTeXt.ico ..... 232
    - NF.ico ..... 232
    - SN ... 235, 238, 582, 583, 593
    - SN.ico ..... 232
    - SN.toaddrindent ..... 583
    - SNleft.ico ..... 232
    - SNmmarg ..... 583
    - toaddrindent ..... 583
  - margin**
    - \marginline ... 143, 229, 230, 314, 523, 526, 539
    - \marginnote .. 523, 524, 539
    - \marginparsep ..... 131
    - \marginparset ..... 530
    - \marginparwidth .... 24, 131
    - \marginpar 22, 142, 143, 229, 314, 523, 526-528, 537, 539
    - \mise en page
      - marginparwidth ..... 537
    - \normalmarginpar . 527, 528, 537, 538
    - \reversemarginpar . 527, 528, 537, 538
  - marginline ..... 142, 229
  - marginnote ..... 523, 538-540
  - marginpar . 272, 526-528, 531, 533, 535, 537, 538, 540, 541
  - normalmarginpar .. 524, 537

- reversemarginpar ... 524, 538
- textwithmarginpar ..... 272
- mark** ..... 103, 504, 506, 509
  - \GenericMarkFormat ... 504, 509-511
  - \Paragraphmark ..... 330
  - \automark\* ... 249, 261-263, 265
  - \automark\*\* ..... 499
  - \automark 249, 250, 261-265, 268, 499, 500, 503, 506, 513
  - \botmark ..... 505, 512
  - \chaptermarkformat 104, 511
  - \deffootnotemark .... 79, 80, 223, 224, 309, 310
  - \deffootnote ..... 223, 309
  - \firstmark ..... 505, 512
  - \footnotemark ... 76-78, 222, 223, 308
  - \headmark ... 267, 501, 502, 520
  - \leftbotmark ..... 505, 512
  - \leftfirstmark . 267, 502, 505, 512
  - \leftmark . 267, 268, 501-505, 511, 512, 520
  - \lefttopmark ..... 505, 512
  - \letterpagemark .... 591, 592
  - \listremarkboxnames ... 457
  - \manualmark . 249, 250, 261, 262, 265, 268, 499, 500, 503
  - \markboth ..... 68, 70, 217, 218, 262, 268-270, 330, 362, 503-506, 512, 513, 590, 610
  - \markleft . 70, 217, 218, 262, 268, 503, 506, 590
  - \markright . 68, 70, 217, 218, 262, 268, 330, 362, 503, 505, 506, 512, 590, 610
  - \pagemark ... 252, 267, 501, 502, 591
  - \paragraphmarkformat . 502
  - \paragraphmark ..... 503
  - \partmark ..... 268, 503
  - \rightbotmark 267, 502, 505, 512
  - \rightfirstmark ..... 505, 512
  - \rightmark .... 72, 267, 268, 501-505, 511, 512, 520
  - \righttopmark ..... 505, 512
  - \subparagraphmarkformat ... 502
  - \subparagraphmark ..... 503
  - \thefootnotemark ... 79, 223, 224, 309
  - \topmark ..... 505, 512
  - nommark* ..... 554
  - nommarkformat* ..... 554
  - nommark* ..... 554
  - automark 249, 250, 261, 264, 265, 472, 499, 500, 506, 513
  - automark\* ..... 249, 499, 500
  - chaptermarkformat ..... 330
  - clausemark ..... 362
  - clausemark=
    - forceboth ..... 371
  - clausemark=réglage ..... 361
  - firstmark ..... 505
  - foldmark ..... 176, 180
  - foldmarks 178, 179, 580, 581, 593
  - foldmarks=selection .... 178
  - footnotereference ... 224, 310
  - leftmark ..... 262
  - lfoldmarklength ..... 580
  - lfoldmarkpos ..... 580
  - manualmark .. 249, 250, 261, 264, 265, 499, 500

- 
- marcase=valeur ..... 265
  - markboth ..... 217, 330
  - markcase ..... 266, 501
  - markright ..... 217, 330
  - Paragraphmark ..... 330
  - paragraphmark .... 319, 330
  - remarkbox ..... 457
  - remarkboxname ..... 457
  - Spitzmarke ..... 566
  - thefootnotemark ..... 79
  - Miktex** ..... xxix, xxxiv-xxxvi
    - Options ..... xxxiv
  - minisec** ..... 47
  - mise en page** xv, xvi, xviii, 1, 2,
    - 5-9, 12, 13, 15-21, 23-29, 33,
    - 34, 38, 40, 41, 43, 50, 55, 63,
    - 64, 67, 68, 70, 73, 74, 105,
    - 136, 159-161, 182, 195, 212,
    - 213, 216, 219, 220, 222, 236,
    - 254, 256, 258, 260-262, 264,
    - 270, 271, 273, 291, 292, 297,
    - 298, 301, 303, 304, 309, 314,
    - 362, 449, 479, 504, 512, 520,
    - 523, 524, 527, 531, 545, 546,
    - 568, 569, 601
  - @everystyle@ . 248, 492, 493,
    - 499, 538
  - \titlehead ..... 54
  - \LastTOCLevelWasHigher ...
    - 449
  - \LastTOCLevelWasSame ....
    - 449
  - \adrrchar ..... 242, 243
  - \and ..... 52, 53, 299, 300
  - \author . 45, 53, 54, 299, 300
  - \centering ..... 81, 230
  - \chapter ..... 45, 46, 434
  - \contentsname ..... 421
  - \date ..... 299
  - \dictumauthorformat ... 107
  - \dictumwidth .. 107-109, 311
  - \g@addto@macro ..... 451
  - \ifVtex ..... 378, 379, 394
  - \ifundefinedorrelax 378, 379,
    - 394, 436
  - \include ..... 405
  - \lastpenalty ..... 448
  - \linespread ..... 5, 34
  - \lipsum ..... 258
  - \maketitle ..... 299
  - \nobreakspace ..... 78, 308
  - \noindent ..... 57
  - \normalsize ..... 81, 310
  - \optiontohead ..... 87
  - \ps ..... 169
  - \publishers ..... 299
  - \raggeddictumauthor ... 107,
    - 108, 311
  - \raggeddictumtext 107, 108,
    - 311
  - \raggeddictum 107, 108, 311
  - \raggedleft 81, 107, 108, 311,
    - 312
  - \raggedright ... 81, 108, 171,
    - 200, 230, 311, 550
  - \relax 81, 394, 472, 485, 496,
    - 508, 535, 540, 541, 549, 551,
    - 590, 633
  - \scr@ifVtex ..... 394
  - \scr@ifundefinedorrelax 394,
    - 451
  - \scr@startsection .. 567, 568
  - \secdef ..... 568
  - \section ..... 434
  - \setchapterpreamble 105-108
  - \setpartpreamble .. 105, 107,
    - 108
  - \subject ..... 299

`\subtitle` ..... 299  
`\syncwithnotecolumns` .. 534,  
 540, 541  
`\syncwithnotecolumn` .. 532,  
 533, 540, 541  
`\textheight` ..... 467, 545  
`\textwidth` ... 107, 108, 311,  
 467, 545  
`\title` ..... 299  
`\tocbasic@addxcontentsline` ..  
 430, 452  
`\newkomavar**` ..... 589  
`\newkomavar` ..... 589  
`hdvips.def` ..... 408  
`linktocpage` ..... 408  
`aftername` ..... 183  
`author` ..... 45, 54  
`chapterentrypagenumber` . 46  
`comaketitle` ..... 45  
`dictumauthorformat` .... 311  
`disposition` ..... 93  
`entête` ..... 198  
`figure` ..... 45  
`footinclude` ..... 23  
`fromalign` ..... 200  
`headinclude` ..... 23  
`HOffset` ..... 595  
`linktocpage` ..... 407  
`mpinclude` ..... 23, 24  
`place` ..... 204  
`placeanddate` ..... 204  
`placeseparatot` ..... 204  
`PPdatamatrix` ..... 199  
`PPheadheight` ..... 199  
`PPheadwidth` ..... 199  
`quotation` ..... 227  
`quote` ..... 227  
`scrhack` ..... 467  
`thechapter` ..... 100

`toname` ..... 177  
`VOffset` ..... 595

**N**

**name** ..... 456  
`\nomnumdepth` ..... 554  
`\abstractname` ..... 400  
`\alsoname` ..... 400, 622  
`\bankname` ..... 599, 600  
`\bibname` ..... 400  
`\ccname*` ..... 599  
`\ccname` ..... 400, 600  
`\chaptername` ..... 400  
`\contentsname` ..... 400  
`\customername` .... 599, 600  
`\datename` ..... 599, 600  
`\emailname` ..... 599, 600  
`\enclname*` ..... 599  
`\enclname` ..... 400, 600  
`\faxname` ..... 599, 600  
`\figurename` ..... 400  
`\glossaryname` ..... 400  
`\headfromname` .... 599, 600  
`\headtoname*` ..... 599  
`\headtoname` ..... 400, 600  
`\indexname` ..... 400  
`\invoicename` ..... 599, 600  
`\listfigurename` ..... 400  
`\listtablename` ..... 400  
`\mobilephonenumber` ..... 600  
`\myrefname` ..... 599, 600  
`\pagename*` ..... 599  
`\pagename` ..... 400, 600  
`\partname` ..... 400  
`\phonename` ..... 599, 600  
`\prefacename` ..... 400  
`\proofname` ..... 400  
`\refname` ..... 400

- 
- `\seename` ..... 400
  - `\subjectname` . 208, 599, 600
  - `\tablename` ..... 400
  - `\wwwname` ... 189, 599, 600
  - `\yourmailname` .... 599, 600
  - `\yourrefname` ..... 599, 600
  - `emailname` ..... 189
  - `emailseparator` ..... 188, 189
  - `faxname` ..... 189
  - `faxseparator` ..... 188, 189
  - `headfromname` ..... 188
  - `indexname` ..... 627
  - `mobilephonenumber` ..... 189
  - `mobilephoneseparator` .. 188, 189
  - `mobilphonenumber` ..... 189
  - `pagename` ..... 591
  - `parcitename` ..... 331
  - `phonenumber` ..... 189
  - `phoneseparator` .... 188, 189
  - `sentencesitename` ..... 331
  - `urlseparator` ..... 188, 189
  - newcounter** ..... 554
  - newkomavar** ..... 589
  - newkomavar\*** ..... 589
  - newopt** ..... 412
  - nexfoo** ..... 412
  - ngerman** ..... 597
  - no** ..... 20, 199
  - nobabel** ..... 420, 434, 450
  - nobreakspace** ..... 362
  - noindent** ..... 116, 227, 546
  - nomenclature**
    - `hack` ..... xxii, 31, 468
    - `nlo` ..... 468
    - `nls` ..... 468
    - `nomencl` ..... 468
    - `thenomenclature` ..... 468
  - note** ... xxvii, 46, 53, 68, 76-81, 122, 142, 156, 161, 164, 168, 171, 176, 184, 192, 196, 209, 210, 222-224, 229, 230, 271, 272, 307-310, 314, 380, 504, 505, 510, 512, 523, 526-541, 574, 597
    - `\ProvideNewNoteColumn` .... 539
    - `\ProvideNoteColumn` ... 527, 537
    - `\footnote` ..... 78, 308
    - `\label` ..... 223, 308
    - `\makenote*` ..... 531
    - `\makenote` ..... 531-533
    - `\nofiles` ..... 532
    - `\raggedfootnote` . 80, 81, 224
    - `makenote` ..... 539, 540
    - `note column name` ..... 527
    - `note de bordure` ..... 142
    - `note de marge` ..... 229, 314
    - `note de marge gauche` ... 229
    - `note de pied de page` 52, 299
  - noupper** ..... 501
  - number** ..... 320, 336, 360, 548
    - `\ifnumbered` ..... 105
    - `\ifparnumber` ..... 327, 343
    - `\ifunnumbered` ..... 105
    - `\nonumberline` ..... 452
    - `\numberbox` ..... 510, 511
    - `\numberline` ..... 318, 435, 437-440, 442, 443, 452
    - `\numbertrue` ..... 321
    - `\pagenumbering` .. 72, 73, 83
    - `\parnumberfalse` ... 319, 321, 324, 327
    - `\parnumbertrue` ... 319, 324, 327
    - `\parshortname` ..... 353

`\thisparnumber` ..... 345  
`\todayspanumber` ..... 277  
`\usetocbasicnumberline` 437,  
 438  
`bibnumbered` ..... 59  
`breakafternumber` .. 440, 442  
`entrynumberformat` 442, 443  
`ifnumbered` ..... 397  
`indentunnumbered` .. 60, 141  
`index=numbered` ..... 148  
`leftalignunnumbered` .... 141  
`listofnumbered` ..... 60  
`nonumberline` ... 60, 141, 435,  
 452  
`notnumbered` ..... 142, 149  
`numberbox` ..... 440, 445  
`numbered` 141, 149, 420, 435,  
 437  
`numberedbib` ..... 59  
`numberedlistof` ..... 60  
`numberedtoc` ..... 141  
`numberformat` ..... 442  
`numberline` 60, 141, 435, 442,  
 443  
`numbers=cadre` ..... 88  
`numbers=endperiod` ..... 88  
`numbers=noendperiod` ... 88  
`numsep` ..... 440  
`numwidth` ..... 440  
`pagenumber` .. 47, 69, 70, 177,  
 216-218, 252, 267, 502, 591  
`pagenumber=emplacement` ...  
 216  
`pagenumberformat` . 440, 445  
`parnumber` ..... 329  
`parnumber=auto` ..... 361  
`parnumber=false` ..... 361  
`parnumber=manuel` .... 361  
`parnumber=réglage` ..... 361

`parnumber=true` ..... 361  
`partnumber` ..... 560  
`raggedpagenumber` 440, 444,  
 445  
`symbolinnumbers` ..... 189  
**number=** ..... 320  
**numeric**  
   `parnumeric` ..... 367  
   `sentencenumeric` ..... 367  
**numsep** ..... 442, 444  
**numwidth** .. 438, 439, 442-444,  
 458

**O**

**off** ..... 20, 199  
**offset** ..... 480  
**oldfoo** ..... 412  
**oldopt** ..... 412  
**on** ..... 20, 186  
**onstarthigherlevel** .. 440, 444,  
 449  
**onstartlowerlevel** ... 440, 444,  
 449  
**onstartsamelevel** 440, 445, 449  
**orthotypographie** 64, 320, 636

**P**

**package** ..... xvi, 693  
   `\RequirePackageWithOptions`  
     378, 411, 413  
   `\RequirePackage` .. 378, 405,  
     408, 411, 413, 414  
   `\usepackage` .....  
     9, 10, 13, 14, 16, 21, 38, 39,  
     41, 158, 161, 235, 255, 282,  
     290, 291, 293, 297, 317, 319,  
     357, 378, 405, 408, 411, 413,  
     414, 419, 453, 464

- 
- page** ..... 458  
**pagefoot** ..... 47  
**pagehead** ..... 258  
**pageheadfoot** ..... 258  
**paq-liste** ..... 9  
**par**  
    \ManualPar ..... 327, 341  
    \Par ..... 327  
    \getmaxpar ..... 327, 355  
    \l@cpa ..... 343  
    \parbox ... 53, 107, 129, 195,  
        218, 300, 311, 539  
    \parindent ..... 450  
**paragraph** ... 47, 105, 249, 262,  
        320, 361, 362, 444, 499, 548  
    \AutoPar ..... 327, 341  
    \contract@paragraph@font ..  
        337  
    \contract@paragraph ... 339  
    \contract@subparagraph 339  
    \minisec ..... 46  
    \paragraphe ..... 320  
    \paragraphformat ..... 100  
    \paragraphmarkformat . 267  
    \paragraphmark ..... 268  
    \paragraphnumdepth ... 104  
    \paragraph ... 335, 358, 554,  
        555, 557  
    \parciteformat ..... 327  
    \parformat ..... 327, 343  
    \p@par ..... 343  
    \raggedpart ..... 100  
    \raggedsection ..... 100  
    \sectionnumdepth ..... 104  
    \section ..... 47, 472, 553  
    \subparagraph\* ..... 96  
    \subparagraphformat ... 100  
    \subparagraphmarkformat ...  
        267  
    \subparagraphmark ..... 268  
    \subparagraphnumdepth .....  
        104, 105  
    \subparagraph ... 46, 48, 90,  
        99, 361, 472, 553  
    \subsectionnumdepth ... 104  
    \subsection . 48, 62, 63, 104,  
        356, 357, 472, 553, 561  
    \subsubparagraph ..... 472  
    \subsubsectionnumdepth 104  
    \subsubsection .. 48, 62, 104,  
        472, 553, 554, 556  
    \theHpar ..... 343  
    \theparagraph ..... 101  
    \thepar .. 327, 343, 344, 361,  
        362  
    \thesubparagraph ..... 101  
arabicpar ..... 361  
chapitre ..... 443, 448  
chapter ... 105, 249, 262, 499  
mparhack ..... 523  
nopar ..... 319, 323  
nosentence ..... 323  
pagination ..... 47  
par ... 327, 343, 361, 539, 565  
Par. .... 368  
par. .... 369  
paragraph  
    \paragraph\* ..... 96  
    \paragraph .. 90, 361, 472,  
        553  
paragraphe .....  
    xxxiv, 3, 7, 41, 42, 45, 53,  
    57, 62-66, 75, 79, 100, 104,  
    106, 109, 114, 116, 118, 119,  
    129, 145, 147, 155, 160, 164,  
    168, 169, 172, 212, 213, 223,  
    227, 228, 230, 246, 262, 293,  
    295, 299, 308, 309, 313, 314,

- 317-329, 333-337, 339, 341-347, 352, 356-374, 427, 435, 443, 448, 470, 473, 499, 504, 516, 524, 527, 529, 531-534, 536, 546, 552, 557-559, 565, 566, 574, 606, 612, 614-616, 619
- paragraphe 3.13 ..... 604
- parlong ..... 319
- parnumeric ..... 319
- parshort ..... 319
- part .. 95, 105, 249, 262, 437, 439, 442-444, 448, 499, 500, 559, 560
- phrase ..... 346
- secnumdepth ..... 104
- section 47, 105, 249, 262, 319, 443, 448, 499, 500
- sectionentry ..... 47
- sectionentrypagenumber .. 47
- sous-paragraphe ... 262, 320, 324, 325, 339, 346, 347, 356, 373, 473
- sous-section ..... 262
- sous-sous-section ..... 262
- subparagraph ... 48, 105, 249, 262, 320, 438, 443, 499, 558
- subparagraphformat .... 104
- subsection 105, 249, 250, 262, 437, 439, 443, 448, 499, 500, 548, 561
- subsubsection . 105, 262, 437, 443, 499, 548, 556
- parbox** ..... 108, 564
- parellipsis** .. 321, 333, 362, 363
- paroff** ..... 367
- part** ..... 553
  - \addpart\* ..... 96
  - \addpart ..... 97
  - \part\* ..... 96, 97, 467
  - \partformat ..... 100
  - \partmarkformat ... 267, 502
  - \partmark ..... 268, 503
  - \partnumdepth ..... 104
  - \partpagestyle ..... 96, 97
  - \part ..... 46, 47, 62, 88, 90, 92, 93, 96, 97, 99-102, 104, 105, 138, 263, 434, 472, 553, 560, 562, 563
  - \raggedpart ..... 99
  - part ..... 47
  - partentry ..... 47, 63
  - partentrypagenumber 47, 62, 63
  - partnumber ..... 47
- partie** ..... 548
- PDF** . xxxi, 31, 32, 394, 407, 545
- pdf**
  - \pdfflastypos ..... 531
  - \pdf ..... 408
  - \pdfpageheight .. 31, 32, 531
  - \pdfsavepos ..... 531
  - hpdf.tex.def ..... 408
- PDK** ..... xxxi
- police**
  - \normalfont ..... 44
  - \rmfamily ..... 44
  - \sffamily ..... 44
  - \Huge ..... 44, 525
  - \LARGE ..... 44, 525
  - \Large ..... 44, 525
  - \NormalColor ..... 252
  - \aliaskomafont ..... 550, 551
  - \bfseries 44, 93, 112, 197, 525
  - \big ..... 93
  - \font=attributs de police 528
  - \footnotesize ..... 44, 525
  - \huge ..... 44, 525



- 
- \itshape ..... 44, 525
  - \large ..... 44, 525
  - \mdseries ..... 44, 94, 525
  - \newkomafont ..... 551
  - \normalcolor ... 93, 107, 525
  - \normalfont 53, 63, 107, 252
  - \normalsize ..... 44, 525
  - \normal ..... 93
  - \rmfamily ..... 525
  - \scriptsize ..... 44, 525
  - \scshape ..... 44, 525
  - \setkomafont ..... 44, 45, 52-54, 56, 69, 79-81, 85, 93, 108, 112, 113, 126, 180, 195, 197, 199, 204, 206, 207, 224, 226, 227, 246, 250, 251, 254, 257, 267, 271, 295, 300, 301, 309-313, 358, 502, 525-528, 537, 539, 550, 551, 553, 557-559, 594, 595
  - \sffamily .. 93, 107, 112, 196, 197, 525
  - \slshape ..... 44, 252, 525
  - \small ..... 44, 93, 525
  - \textsuperscript 43, 79, 224, 295, 309, 525
  - \tiny ..... 44, 525
  - \ttfamily ..... 44, 525
  - \upshape ..... 44, 525
  - \useencodingofkomafont 251, 296, 526
  - \useencodinguikomafont . 48
  - \usefamilyofkomafont ..... 48
  - \usefontofkomafont .. 251, 296, 526
  - \usefontkomafont ..... 48
  - \usefontofkomafont . 48, 251, 296, 526, 551
  - \usekomafontpagenumber .... 252
  - \usekomafont ..... 44, 45, 48, 53, 54, 63, 70, 246, 250, 251, 295, 296, 525, 526, 550, 551
  - \useseriesofkomafont 48, 251, 296, 526
  - \useshapeofkomafont .... 48, 251, 526
  - \useshapofkomafont .... 296
  - \usesizeofkomafont . 48, 251, 296, 526
  - normalfont ..... 252
  - declaration ..... 537
  - encoding ..... 251
  - family ..... 251
  - font ..... 553, 556-562
  - fontseries ..... 251
  - fontshape ..... 251
  - Latin Modern ..... 530
  - normalfont ..... 252
  - prefixfont ..... 558-560
  - priority ..... 197
  - setkomafont ..... 45, 176
  - textsuperscript ..... 79
  - usesizeofkomafont ..... 251
  - vfont ..... 560
  - police\bfseries** ..... 440
  - Port-Payé** ..... 197
  - PP** ..... 584
  - PS** ..... xxxi, 169, 407
  - pseudo-longueur**
    - PLengthspecialmailindent .... 197
    - specialmailrightindent ... 197
  - publishers** ..... 47

**R**

- ragged2e** ..... 82
- raisebox** ..... 130
- recto-verso** ..... 258
- Redeclare**
- `\RedeclareLayer` ... 476, 486
  - `\RedeclarePageStyleAlias` .... 488
  - `RedeclarePageStyleByLayers` . 489
  - `RedeclareSectionCommand` ... 561
- ref** ..... 154, 331, 365, 368
- `\Sentence` ..... 326
  - `\customer` ..... 204
  - `\dataname` ..... 204
  - `\defaultrefields` ..... 590
  - `\footref` ..... 78, 79, 223, 308
  - `\invoicename` ..... 204
  - `\label` ..... 78, 322, 364
  - `\myref` ..... 204
  - `\pageref` ..... 166, 364
  - `\parname` ..... 326, 353, 369
  - `\parshortname` ..... 369
  - `\refClauseN` ..... 322, 365
  - `\refClause` ..... 322, 327, 365
  - `\refL` ..... 322, 345, 364, 365
  - `\refN` ..... 322, 364, 365
  - `\refParL` . 322, 349, 365, 366
  - `\refParN[arabic]` ..... 322
  - `\refParN` ..... 322, 349, 365
  - `\refParS` ..... 322, 349, 365
  - `\refParagraph` ..... 346
  - `\refPar` .. 322, 323, 347, 365, 367
  - `\refSentenceL` ..... 322, 365
  - `\refSentenceN` ..... 322, 365
  - `\refSentenceS` ..... 322, 365
  - `\refSentenceindex` ..... 367
  - `\refSentence` .. 322, 323, 350, 365, 367
  - `\refS` ..... 322, 364, 365
  - `\refl` ..... 365
  - `\refstepcounter` .... 326, 368
  - `\ref@L` ..... 345
  - `\ref@N` ..... 346
  - `\ref@ParX` ..... 349
  - `\ref@ParagraphN` ..... 347
  - `\ref@Paragraph` ..... 346
  - `\ref@Par` ..... 347
  - `\ref@Sentence` ..... 350
  - `\ref` . 166, 322, 323, 327, 364, 365, 367, 541
  - `\removereefields` ..... 589
  - `\sentencename` ..... 353, 369
  - `\sentenceshortname` 353, 369
  - `\yourmailname` ..... 204
  - `\yourref` ..... 204
  - `\defaultrefields` ..... 589
  - `\removereefields` ..... 589
  - `refPar` ..... 327
  - `ref\ParagraphN` ..... 347
  - `ccseparator` ..... 152
  - `customer` ..... 152, 203
  - `date` ..... 152
  - `emailseparator` ..... 152
  - `enclseparator` ..... 152
  - `faxseparator` ..... 152
  - `firstfoot` ..... 152
  - `firsthead` ..... 152
  - `footnotereference` 46, 80, 176
  - `footref` ..... 223
  - `fromaddress` ..... 152
  - `frombank` ..... 152
  - `fromemail` ..... 153
  - `fromfax` ..... 153
  - `fromlogo` ..... 153

- 
- frommobilephone ..... 153  
fromName ..... 153  
fromphone ..... 153  
fromurl ..... 153  
fromzipcode ..... 153  
invoice ..... 153, 203  
location ..... 153  
longpar ..... 367  
longsentence ..... 367  
mobilephoneseparator ... 153  
myref ..... 153, 203  
nextfoot ..... 153  
nexthead ..... 153  
nopar ..... 367  
nosentence ..... 367  
numeric ..... 367  
numericpar ..... 367  
numericssentence ..... 367  
pageref ..... 541  
ParL ..... 367  
parlong ..... 367  
ParN ..... 367  
parnumeric ..... 367  
paroff ..... 367  
ParS ..... 367  
parshort ..... 367  
phoneseparator ..... 153  
place ..... 153  
placeseparator ..... 153  
ref ... 112, 134, 323, 345, 346  
ref\*{testA} ..... 323  
ref=long ..... 319  
ref=nopar ..... 323  
ref=nosentence ..... 323, 373  
ref=numéric ..... 319  
ref=parlong ..... 319  
ref=réglage ..... 365  
ref=sentenceshort ..... 319  
ref=short ..... 319  
refaftervskip ..... 576  
refClause ..... 368, 373  
refClauseN ..... 368  
refhpos ..... 576, 586  
refL ..... 368  
refline ..... 203, 235, 586  
refline=narrow ..... 586  
refline=selection ..... 203  
refline=wide ..... 203, 586  
refN ..... 368  
refname ..... 177, 204, 206  
refPar ..... 323, 367  
refParL ..... 368  
refParN ..... 368  
refParS ..... 368  
refS ..... 346, 368  
refSentence ..... 323, 327  
refSentenceL ..... 368  
refSentenceN ..... 368  
refSentenceS ..... 368  
refvalue ..... 177, 204, 206  
refvpos ..... 576, 585  
refwidth ..... 576, 586  
ref@S ..... 346  
SentenceL ..... 367  
sentencelong ..... 367  
SentenceN ..... 367  
sentencename ..... 326  
sentencenumeric ..... 367  
SentenceS ..... 367  
short ..... 367  
shortpar ..... 367  
signature ..... 154  
subject ..... 154  
subjectseparator ..... 154  
toaddress ..... 154  
toname ..... 154  
yourmail ..... 154, 203  
yourref ..... 154, 203

- 
- refline** ..... 594  
**remerciements**  
    
**Renouvo** ..... 636  
**replace**  
    \ReplaceClass ..... 411  
    \ReplaceInput ..... 411  
    \ReplacePackage ... 411, 413  
    \UnReplaceClass ..... 413  
    \UnReplaceInput ..... 413  
    \UnReplacePackage ..... 413  
    ReplaceClass ..... 411  
    ReplaceInput ..... 410, 411  
    ReplacePackage ... 411, 414  
**report** .. 37, 84, 92, 96, 262, 289,  
    438, 440, 511  
**Rouquette** ..... xv  
**rule**  
    \dictumrule ..... 108, 311  
    \footnoterule ..... 224  
    \mise en page  
        dictumrule ..... 107  
    \rule ..... xxx, 522  
    \setfootnoterule ..... 310  
    dictumrule ..... 311  
    footnoterule 46, 81, 176, 224,  
        310  
    fromrule .. 176, 183, 185, 186,  
        582  
    fromrule=afteraddress .. 183  
    fromrule=aftername .... 183  
    fromrule=position ..... 183  
    fromrulethickness .. 575, 582  
    fromrulewidth . 183, 575, 582  
    ruled ..... 133  
    setfootnoterule ..... 81  
**répertoire** ..... xxxiv  
    \KOMA-Script ..... xxxvi  
    \LaTeX ..... xxxvi  
    \TeX ..... xxxvi  
    MikTeX-Options ..... xxxvi
- S
- scraddr** ..... 1  
**scrartcl** xxvii, 37, 38, 43, 45, 47,  
    61-63, 68, 71, 84, 85, 92, 96,  
    97, 100, 103, 138, 143, 145,  
    228, 255, 258, 262, 289, 290,  
    357, 370, 372, 472, 547, 552,  
    553, 555, 563, 569  
**scrbase** .... 2, 39, 158, 236, 289,  
    377-380, 383, 384, 386, 388,  
    392, 393, 397, 399-401, 436,  
    454, 506  
    scrbase.sty ..... 379  
**scrbook** ..... xxvii, 13,  
    37, 38, 43, 45, 49, 57, 59-62,  
    68, 71, 82-85, 92, 93, 95-97,  
    100, 103, 104, 106, 138, 139,  
    145, 228, 289, 290, 297, 406,  
    407, 427, 435, 448, 472, 547,  
    552, 555, 559-562, 568, 569  
**scrdate** ..... 1, 275, 278, 279  
**scrxend** 79, 80, 289, 290, 292,  
    293, 295-298, 303, 305, 306,  
    309, 312, 547, 550  
    \scrxend ..... 300, 309  
**scrguide** ..... xx-  
    viii  
**Script** ..... xx-  
    viii  
**Script2** ..... xx-  
    viii  
**scrjura** . 142, 315-320, 324, 325,  
    333, 356-358, 361, 362, 364,  
    366, 369-371, 375  
    \scrjura@lang@error ... 353

- 
- `\scrjura@special@par` .. 345
  - `\scrjura@special@reset@par` . 345
  - `scrjura.dtx` ..... 316
  - `scrjura.sty` ..... 321
  - scrkbase** ..... 317, 328, 335
  - scrlayer** .... 245, 248, 249, 261, 262, 265-269, 467, 469, 470, 472-474, 486, 487, 492, 494, 498-507, 510, 512, 513, 521, 524, 527, 531, 537, 538
  - `\InitInterface` ..... 507, 508
  - `\OnAutoRemoveInterface` .... 486, 489, 498
  - `\scrlayer-scrpage` ..... 103
  - `\scrlayerAddCsToInterface` ... 507, 508
  - `\scrlayerAddToInterface` 507, 508
  - `\scrlayerInitInterface` .... 507
  - `\scrlayerOnAutoRemoveInterface` ..... 508
  - `\scrlayer` ..... 498
  - note ..... 527
  - notecolumn ... 523, 524, 530, 531, 536, 537, 539
  - scrpage ... 26, 47, 68-71, 216-218, 245, 246, 248, 251-253, 259-261, 263, 265, 267, 269, 271, 273, 498, 502, 504, 509, 514-516, 519, 520, 522, 524, 527, 530, 531, 591
  - scrletter** ..... 215, 573, 590-592
  - scrlettr** ..... 237, 289
  - dvi ..... 600
  - scrfile** .. 236, 290, 403-405, 408, 410, 413, 414
  - scrlogo.dtx** ..... 316
  - scrlettr2** xx, 1, 37, 151, 152, 156, 157, 159, 160, 162, 164, 165, 167, 169, 174, 176, 182, 185, 186, 195, 197, 198, 200, 201, 203, 206, 207, 209, 210, 212, 215-219, 221, 228, 229, 232, 233, 237, 239, 240, 242, 289, 573, 574, 578, 581, 584, 589, 591, 592, 596, 597, 599
  - scrlettr2.cmd.KOMAOption ... 160
  - scrpage-scrlayer** ..... 258
  - scrpage2** xx, 26, 47, 68-70, 177, 216-218, 245, 249, 253, 261, 262, 273, 319, 411, 499, 509, 516, 519, 522, 590, 591
  - `\scrpage2` ..... 103
  - scrpage2.sty ..... 411
  - scrreprt** ..... xxvii, 37, 38, 43, 45, 49, 57, 59-62, 68, 83-85, 92, 93, 95-97, 100, 103, 104, 138, 139, 145, 228, 289, 290, 427, 435, 448, 472, 547, 552, 555, 559-562, 568, 569, 610
  - scrwfile** 417-419, 421, 422, 432, 434, 450
  - secnumdepth** ... 103, 268, 503, 556
  - `\secnumdepth` ..... 62, 104
  - section** ..... xxvii, xxxiv, 1, 8, 9, 12-15, 19-23, 25, 26, 29, 34, 38-40, 42, 43, 45-47, 51-54, 56-58, 60-62, 64, 65, 68, 69, 71-74, 79-85, 88-90, 92-94, 97, 100, 102, 103, 106-108, 112, 119, 120, 122-126, 138, 140, 143-145, 149, 152-154, 157, 159-162, 168, 174, 176, 177, 179,

195, 197, 200, 212-217, 219-	
230, 232, 235, 246-251, 254,	
256, 257, 261-265, 267-272,	
281-283, 286, 290, 291, 293-	
304, 306, 309-314, 320, 321,	
325-327, 335, 337, 341, 356,	
358, 359, 361, 368-371, 379,	
392, 393, 398, 414, 420, 421,	
426, 428, 429, 432, 436-440,	
442, 444, 448, 455, 460, 464-	
466, 470, 471, 473, 479, 480,	
483, 485-487, 489, 491-493,	
495, 498, 500-502, 504-510,	
512, 514-516, 518, 520, 523,	
525, 526, 528, 531, 532, 534,	
536, 538, 539, 543-545, 548-	
557, 561, 564, 566-569, 573-	
578, 580-582, 584-586, 588-	
592, 597, 598, 600, 605, 607,	
610, 612-614	
<code>\section*</code> 82, 96, 97, 610, 613	
<code>\sectioncatchphraseformat</code> ...	
102, 566	
<code>\sectionlinesformat</code> 102, 566,	
567	
<code>\sectionmarkformat</code> .... 103,	
104, 267, 502, 504, 510, 511	
<code>\sectionmark</code> .. 103, 268, 503	
<code>\section</code> ..... 90	
<code>\subsection*</code> ..... 96, 610	
<code>\subsectionmarkformat</code> . 103,	
104, 267, 502, 504, 510, 511	
<code>\subsectionmark</code> ... 103, 268,	
503	
<code>\subsection</code> ..... 90	
<code>\subsubsection*</code> ..... 96	
<code>\subsubsectionmarkformat</code> ...	
267, 502	
<code>\subsubsectionmark</code> 268, 503	
<code>\subsubsection</code> ..... 90	
<code>\textsectionmarkformat</code> . 103	
<code>\textsubsectionmarkformat</code> ...	
103	
<code>\thesection</code> ..... 101	
<code>\thesubsection</code> ..... 101	
<code>\thesubsubsection</code> ..... 101	
section 2.9 ..... 145	
sectionentry ..... 63	
sectionentrydotfill ..... 61	
sectionentrydots .. 47, 61, 63	
sectionentryfill ..... 61	
sectionentrypagenumber . 47,	
62, 63	
sectionentrywithdots ..... 61	
sectionentrywithoutdots .. 61	
sectionmark .... 103, 505, 512	
sous-section .. 100, 338, 504,	
613	
subsectionmark ..... 103	
<b>section 2.9</b> ..... 47, 90, 105	
<b>sectioning</b> ..... 48	
<b>sentence</b>	
nosentence ..... 319	
sentencelong ..... 319	
sentenceoff ..... 367	
sentenceshort ..... 319	
sentencesnumeric ..... 319	
<b>set</b>	
<code>\Setparpreamble</code> ..... 92	
<code>\setkomavar*</code> ..... 154	
<code>\setkomavar</code> .. 154, 171, 172,	
183, 188, 192, 195, 196, 200,	
201, 203, 204, 206, 212, 218,	
230, 591	
<code>\setparsizes</code> ..... 552	
setspace .... 5, 17, 18, 34, 467	
<b>short</b>	
<code>\defineshorthand</code> ..... 364	

- 
- \usesorthand ..... 364
  - parshort ..... 367
  - sentenceshort ..... 367
  - shortsentence ..... 367
  - show**
  - \showISOenvelope ..... 595
  - \showUScheck ..... 595
  - \showUScommercial .... 595
  - \showenvelope ..... 595
  - side**
  - autooneside ..... 265
  - twoside . 20, 42, 74, 220, 258,  
    304, 381, 479, 482
  - twoside.floatpage ..... 479
  - twoside=
  - semi ..... 20, 21
  - valeur marche/arrêt .... 20
  - signature** ... 168, 184, 185, 230
  - skip** ..... 573
  - \@tempkipa ..... 562, 568
  - \afterskip ..... 555
  - \baselineskip 17, 25, 27, 562
  - \bigskip . 109, 114, 116, 146,  
    227, 534
  - \chapterheadendvskip .. 562,  
    563
  - \chapterheadmidvskip .. 562
  - \chapterheadstartvskip . 106,  
    562, 563
  - \enskip ..... 100
  - \footskip . 247, 248, 479, 588
  - \medskip ..... 114, 227
  - \newdimen ..... 480, 483
  - \newlength ..... 480, 483
  - \newskip ..... 480, 483
  - \parfillskip ..... 450
  - \parskip .. 435, 450, 560-562
  - \postskip ..... 321
  - \preskip ..... 321
  - \skip ..... 395
  - \topskip ..... 17, 28
  - \vskip ..... 442
  - afterskip .. 553, 554, 556-562
  - align ..... 480
  - beforeskip 440, 442, 553, 554,  
    556-562
  - footskip ..... 479, 498
  - ifisskip ..... 396
  - innerskip ..... 558-560
  - noparskipfake ..... 435
  - parskip . 64, 65, 120, 212, 361
  - parskip=
  - false ..... 66, 213
  - half ..... 370
  - méthode ..... 64
  - relative ..... 552
  - postskip ... 321, 336, 358, 360
  - PPdatamatrixvskip 199, 584
  - preskip .... 321, 336, 358, 360
  - refaftervskip ..... 586
  - sigbeforevskip . 230, 576, 587
  - smallskip ..... 284
  - subjectaftervskip .. 576, 586,  
    587
  - subjectbeforevskip . 577, 586,  
    587
  - topskip ..... 15
  - split** ..... 622
  - standard** ..... xxv, xxx, 2, 29,  
  30, 37, 42, 43, 49-51, 55, 57,  
  62, 63, 68, 69, 72, 74, 77, 79,  
  84, 85, 90, 92, 96, 99, 105,  
  109, 114, 116, 122, 125, 129,  
  135, 139, 149, 174, 220, 222,  
  223, 225, 227, 245-249, 253,  
  255, 260-262, 265, 268, 289,  
  293, 294, 297, 298, 301, 304,  
  307, 309, 312, 313, 319, 404,

- 437-440, 442-444, 454, 458,  
472, 473, 500, 501, 503, 504,  
506, 511-513, 543, 547, 548,  
569, 597, 601, 608, 622
- style de page**
- \DeclareAliasPageStyle . 491
- \DeclareNewPageStyleBy-  
Layers ..... 491
- \DeclarePageStyleByLayers ..  
491
- \DestroyAliasPageStyle . 508
- \DestroyLayer ..... 485, 508
- \DestroyRealLayerPageStyle .  
497, 508
- \ForEachLayerOfPageStyle\* .  
493, 494
- \ForEachLayerOfPageStyle ...  
492-494
- \ModifyLayerPageStyleOp-  
tions ..... 493, 496
- \afterheading ..... 341
- \autoclearnotecolumns .. 540
- \cfoot ..... 259
- \chapterpagestyle .... 93, 96
- \chead ..... 522
- \cheah ..... 259
- \cleardoubleemptypage .....  
74, 75, 84, 86, 87, 221, 304,  
305
- \cleardoubleevenemptypage ..  
74, 75, 304, 305
- \cleardoubleevenpageusingstyle  
74, 75, 304, 305
- \cleardoubleevenpage .... 74,  
220, 304
- \cleardoubleevenplainpage ...  
74, 75, 304, 305
- \cleardoubleevenstandardpage  
74, 75, 304
- \cleardoubleoddemptypage ...  
74, 220, 304, 305
- \cleardoubleoddpageusingstyle  
74, 75, 220, 304, 305
- \cleardoubleoddpage 74, 220,  
221, 303, 304, 306, 546
- \cleardoubleoddplainpage ....  
74, 220, 304, 305
- \cleardoubleoddstandardpage  
74, 75, 220, 304, 305
- \cleardoublepage=  
current ..... 303
- \cleardoublepageusingstyle ...  
74, 84, 86, 87, 221, 304, 305
- \cleardoublepage ..... 69,  
73-75, 84, 86, 87, 220, 221,  
303-306, 546, 610
- \cleardoubleplainpage 74, 75,  
84, 86, 87, 221, 304, 305
- \cleardoublestandardpage ....  
74, 75, 84, 86, 87, 221, 304,  
305
- \clearnotecolumns . 536, 540
- \clearnotecolumn .. 536, 540
- \clearpage ..... 73, 74,  
86, 220, 304, 409, 536, 540,  
546, 610, 613, 633
- \clearscrheadfoot ..... 516
- \clearscrheadings ..... 515
- \clearscrplain ..... 515
- \deftocheading ..... 434-437
- \iffalse ..... 472
- \ifoot ..... 259
- \iftrue ..... 472
- \if@chapter ..... 472
- \item ..... 46
- \l@Eintragsebene ..... 440
- \maketitle ..... 47, 48, 50
- \nextfoot ..... 218



- 
- \nexthead ..... 218
  - \ofoot ..... 259
  - \ohead ..... 259
  - \onecolumn ... 435, 610, 633
  - \partpagestyle ..... 93
  - \publishers ..... 47
  - \rhead ..... 259
  - \section ..... 47
  - \subject ..... 48
  - \subtitle ..... 48
  - \thispagestyleempty ..... 50
  - \thispagestyle ..... 298
  - \titlehead .. 48, 52, 299, 300
  - \titlepage
    - maketitle ..... 298
  - \titlepagestyle ..... 303
  - \titlepage ..... 50, 298, 303
  - \title ..... 48
  - \twocolumn ..... 610, 633
  - \commaatlist ..... 494
  - \edf ..... 494
  - \gedf ..... 494
  - above ..... 124
  - Alias ..... 516
  - autoclearnotecolumns .. 536, 540
  - below ..... 124
  - besideinner ..... 124
  - besideouter ..... 124
  - besideright ..... 124
  - besidetop ..... 125
  - bot ..... 124, 218
  - botcenter ..... 218
  - botcentered ..... 218
  - botleft ..... 218
  - botmiddle ..... 218
  - botright ..... 218
  - bottom ..... 124, 458
  - bottombeside ..... 130
  - cefoot ..... 257, 515
  - cehead ..... 256, 514
  - center ..... 219
  - centered ..... 219
  - centeredbeside ..... 130
  - cfoot ..... 515
  - chapter ..... 95
  - chead ..... 515
  - cleardoubleemptypage ... 83, 221
  - cleardoubleevenemptypage ... 221
  - cleardoubleevenpage .... 221
  - cleardoubleevenpageusingstyle 221
  - cleardoubleevenplainpage ..... 221
  - cleardoubleevenstandardpage . 221
  - cleardoubleoddemptypage .... 221
  - cleardoubleoddpage ..... 221
  - cleardoubleoddpageusingstyle 221
  - cleardoubleoddplainpage 221
  - cleardoubleoddstandardpage . 221
  - cleardoublepage . 73, 83, 220, 221, 303, 306
  - cleardoublepage=
    - empty ..... 305
    - plain ..... 304
  - cleardoublepagestandardpage . 83
  - cleardoublepageusingstile . 83
  - cleardoublepageusingstyle .... 221
  - cleardoubleplainpage 83, 221

- cleardoublestandardpage 83, 221
- clearmainofpairstyles .. 515
- clearpage ..... 83, 221
- clearpairstyles .... 515
- clearplainofpairstyles .. 515
- CloneTOCStyle ..... 438
- cofoot ..... 257, 258, 515
- cohead ..... 256, 514
- codate ..... 46
- codedication ..... 46
- coitem ..... 46
- comaketitle ..... 46
- cominsec ..... 47
- dateright ..... 203
- deactivatepagestylelayers ..... 493, 494
- defpairstyles ..... 515
- dottedtocline ..... 443
- dynamwidth .. 440, 442, 444
- empty ..... 68, 74, 75, 215-217, 220, 245, 248, 263, 298, 305, 492
- entête ..... xvi, xix, 2, 13, 21-27, 29, 41, 47, 49, 52-57, 62, 67-71, 75, 82, 83, 87, 88, 91, 101-103, 105, 106, 113, 114, 161, 167, 176-178, 182-186, 188-190, 192, 195, 197-199, 203, 211, 214-220, 222, 232, 243, 245, 247, 248, 251-262, 264-266, 268-273, 297-301, 303, 305, 318-320, 337, 400, 420, 437, 452, 454, 467, 480, 486, 492, 498, 499, 501, 505, 506, 509, 513, 514, 516-522, 544, 554, 555, 557-559, 573-575, 579, 581-584, 588, 590, 593-596
- false ..... 219
- firsthead ..... 195
- foot ..... 218
- footbotline ..... 520
- footcenter ..... 218
- footcentered ..... 218
- footleft ..... 218
- footmiddle ..... 218
- footright ..... 218
- footsepline ..... 520
- GetRealPageStyle ..... 492
- graduated ..... 60
- hbp ..... 458
- head ..... 219
- headcenter ..... 219
- headcentered ..... 219
- heading ..... 122-124
- headings ..... 68, 69, 86, 95, 103, 153, 216, 218, 245, 249, 253, 260-262, 468, 499, 561, 562, 591, 613
- headings=
  - big ..... 85, 95, 561
  - cadre ..... 85
  - normal ..... 85, 560
  - opelleft ..... 85
  - openany ..... 85
  - openright ..... 85, 86
  - optiontohead ..... 90
  - optiontoheadandtoc .... 90
  - optiontotoc ..... 90, 92
  - small ..... 85
- headings=big ..... 93, 562
- headings=normal ..... 562
- headings=onelinechapter . 85
- headings=small ..... 562
- headings=twolinechapter . 85

- 
- headleft ..... 219
  - headmiddle ..... 219
  - headright ..... 219
  - headsepline ..... 520
  - headtopline ..... 520
  - ifoot ..... 515
  - ihead ..... 515
  - innerbeside ..... 124, 130
  - kolumntitel ..... 468
  - labelinglabel ..... 47
  - largetocline ..... 443
  - tfoot ..... 257, 514
  - left ..... 219
  - leftbeside ..... 124, 130
  - lehead ..... 256, 514
  - letter . 37, 164, 166, 215, 234,  
507, 591, 599
  - tfoot ..... 257, 258, 515
  - lehead ..... 256, 514
  - maketitle ..... 298
  - middle ..... 219
  - myheadings ..... 68-70, 153,  
216-218, 245, 249, 261, 262,  
371, 499
  - narrow ..... 203
  - newpairstyles ..... 515
  - nextfoot .. 177, 217, 218, 590,  
591
  - nexthead . 217, 218, 590, 591
  - niveau ..... 1, 6,  
31, 46-48, 60-63, 68, 81, 82,  
84, 90, 92, 93, 100, 101, 103-  
105, 110, 111, 123, 125, 130,  
138, 141, 145, 147, 225, 226,  
245, 248-250, 262-264, 268,  
310, 318, 356, 420, 422, 429,  
430, 434-449, 452, 454, 459,  
460, 469, 470, 472-476, 478-  
481, 484-486, 492-500, 504,  
509, 520, 522, 527, 548, 549,  
554, 555, 563, 565-567
  - niveaux ..... 19, 43,  
59, 62, 84, 92-94, 96, 97, 99-  
101, 104, 105, 110, 111, 144,  
225, 226, 249, 256, 262, 263,  
266, 268, 312, 320, 429, 437,  
438, 440, 442-445, 449, 469,  
472-474, 476, 479, 486, 489,  
492-494, 497, 499, 500, 503,  
505, 512, 513, 520-522, 527,  
566
  - no ..... 219
  - nom.foot.above.line .... 520
  - nom.foot.below.line .... 520
  - nom.foot.even ..... 520
  - nom.foot.odd ..... 520
  - nom.foot.oneside ..... 520
  - nom.head.above.line .... 520
  - nom.head.below.line .... 520
  - nom.head.even ..... 520
  - nom.head.odd ..... 520
  - nom.head.oneside ..... 520
  - note ..... 46
  - off ..... 219
  - tfoot ..... 515
  - ohead ..... 515
  - onbackground ..... 491
  - onecolumn ..... 420, 612-614
  - onevenpage ..... 491
  - onfloatpage ..... 491
  - onforeground ..... 491
  - oninit ..... 490
  - onnonfloatpage ..... 491
  - onoddpaper ..... 490
  - ononeside ..... 490
  - onselect ..... 490
  - ontwoside ..... 490
  - outerbeside ..... 124, 130



- 
- |   |                  |                               |                             |
|---|------------------|-------------------------------|-----------------------------|
| tableabove .....                          | 125              | bibliographynumbered ...      | 58                          |
| tableheading .....                        | 125              | dottedtocline .....           | 438, 443                    |
| tablesignature .....                      | 124, 125         | flat .....                    | 59, 138, 139                |
| topatable .....                           | 125              | graduated .....               | 59                          |
| <b>TBMPL</b> .....                        | 180              | indentunnumbered .....        | 468                         |
| <b>tcolorbox</b> .....                    | 627              | index .....                   | 58                          |
| <b>TDS-Baum</b> .....                     | xxxiii           | largetocline .....            | 440                         |
| <b>tex</b> .....                          | xx, xxv, xxix    | listof .....                  | 138                         |
| <b>tex live</b> .....                     | xxix             | listofnumbered .....          | 138                         |
| <b>texmf</b> .....                        | xxxiii           | minitoc .....                 | 417                         |
| <b>texte</b> .....                        | xvi              | nolistof .....                | 138                         |
| <b>textellipsis</b> .....                 | 321, 363         | nonnumberline .....           | 435                         |
| <b>thenom</b> .....                       | 554              | nottotoc .....                | 147                         |
| <b>title</b> .....                        | 48               | numberline .....              | 435, 442, 443               |
| <b>titlepage</b> .....                    | 49, 50, 298      | section 2.9 .....             | 142                         |
| false .....                               | 49, 297          | tocbasic .....                | 438, 440, 450               |
| firstiscover ...                          | 49, 55, 297, 301 | tocdepth .....                | 421                         |
| on-off-valeur .....                       | 297              | toclevel .....                | 440                         |
| première de couverture ...                | 49               | tocline .....                 | 438-440, 443                |
| true .....                                | 49, 297          | tocnumbered .....             | 141                         |
| valeur de marche/ arrêt ..                | 49               | totoc .....                   | 142                         |
| <b>toc</b> 58, 59, 61, 62, 142, 145, 149, |                  | totocnumbered ....            | 141, 142,                   |
| 420-422, 429, 430, 440, 449,              |                  | 146, 147, 149                 |                             |
| 450, 452, 460                             |                  | undottedtocline .....         | 443                         |
| \DeclareNewTOC .....                      | 440              | <b>tocbasic</b> .....         | 229, 290, 328,              |
| \TOCclone .....                           | 420              | 419-424, 426, 428, 430, 433,  |                             |
| \setuptoc 420, 431, 434, 442,             |                  | 434, 436-438, 441, 442, 447,  |                             |
| 443, 450, 454, 455, 468                   |                  | 449-455, 460, 465, 466, 468   |                             |
| \starttoc .....                           | 450              | <b>tocbasic@</b> .....        | 449                         |
| \tocbasic@@after@hook 451                 |                  | <b>tocbasic@extend@babel</b>  | 426,                        |
| \tocbasic@@before@hook ...                |                  | 449                           |                             |
| 451                                       |                  | <b>tocdepth</b> ..            | 62, 63, 90, 105, 356,       |
| \tocbasic@addxcontentsline ..             |                  | 437, 439, 446                 |                             |
| 429                                       |                  | <b>tocnumwidth</b> .....      | 458, 553,                   |
| \tocbasic@listhead .....                  | 452              | 556-558, 560, 561             |                             |
| \tocbasic@starttoc .                      | 428, 450         | <b>true</b> ..                | 20, 21, 23, 24, 46, 49, 65, |
| \tocbasic .....                           | 452              | 85, 102, 180, 186, 195, 196,  |                             |
| \tocdepth .....                           | 62               | 198, 265, 297, 381, 386, 494, |                             |
| bibliography .....                        | 59               | 496, 497, 590                 |                             |

**twocolumn** ... 21, 105, 612-614  
     valeur marche/arrêt ..... 21  
**type** ..... 456  
**typeset** ..... 410  
**typographie** ..... xv,  
     xvi, xxiii, xxvi, xxviii, xxxi,  
     1, 2, 32, 34, 35, 68, 75, 94,  
     99, 118, 168, 171, 461, 510,  
     636  
     \MakeMarkcase ..... 266  
     \MakeUppercase ..... 266  
     \markcase=valeur ..... 501  
     \uppercase ..... 266, 501  
     ignoreupper ..... 266  
     ignoreuppercase ..... 266  
     markcase  
         markcase=used ..... 260  
     markcase=noupper . 266, 501  
     markcase=upper .. 261, 265,  
         266, 501  
     markcase=used .... 266, 501  
     noupper ..... 266  
     nouppercase ..... 266

U

**UNIX** ..... xxxiv  
**upper** ..... 261, 265, 266, 501  
**UScommercial9** ..... 232, 238  
**UScommercial9DW** . 232, 239  
**usegeometry** ..... 544  
**usekomavar** 155, 188, 189, 195,  
     208, 218, 230, 590  
**usekomavar\*** 155, 188, 208, 590  
**usenet** ..... xxx  
**Usenet.de** ..... xx  
**usepackage** . 246, 247, 464, 470,  
     471

V

**V** ..... 616  
**v** ..... 616  
**variable** ..... 151, 152, 154-156,  
     168, 171, 172, 177, 183, 185,  
     187, 192, 195-201, 204-208,  
     210-212, 216-218, 230, 453,  
     584, 585, 589, 590, 615  
**verbose** ..... 616  
**version** xv, xvii-xx, xxxvi, xxvii,  
     xxix-xxx1, 11, 19, 20, 23, 29,  
     32-34, 40, 73, 74, 76, 82, 83,  
     89, 92, 94, 96, 97, 114, 118,  
     119, 135, 137, 146, 155, 156,  
     159, 162, 166, 195, 200, 204,  
     210, 212, 218, 237, 243, 245,  
     263, 279, 281, 282, 285, 287,  
     291, 315, 316, 320, 323-325,  
     356, 359, 366, 373, 375, 393,  
     396, 399-401, 405-407, 415,  
     417, 418, 431, 437, 442, 451,  
     463, 464, 467, 484, 494, 500,  
     514, 523, 525, 536, 543, 546,  
     547, 551, 552, 573, 588-590,  
     597, 601-604, 616, 620, 626,  
     635, 636  
     \declaration ..... 366  
     2.09 ..... xxvi, xx-  
         viii  
     2.9 ..... 11, 40, 159, 292  
     2.96a ..... 40  
     2.9t ..... 11, 40, 159, 292  
     3.01b ..... 11  
     3.22 ..... xxi  
     3.22a ..... xxi  
     1.2c ..... 610  
     first ..... 11, 40, 159, 292  
     last ..... 11, 40, 159, 160, 292

v1.2c ..... 601  
valeur ..... 40  
version\* ..... 263  
wert ..... 159, 292

**W**

**Wrapper** ... 377, 378, 382, 549,  
550, 565, 567, 611

**X**

**xcolor** .... 81, 224, 296, 310, 530

**Y**

**yes** ..... 20, 186









Lorsque l'on me rappelle que plus de 270 millions de personnes dans le monde s'expriment dans le même langage que MOLIÈRE, je n'hésite plus et c'est en pensant à nos gens que je me lance dans l'adaptation de KOMA-Script, avec l'envie que mon travail soit amélioré et enrichi par des locuteurs ou experts indigènes qui seront les bienvenus en tous temps. Si par exemple j'ai utilisé *package* à la place de paquet (*Paket*) pour désigner l'ensemble cohérent d'un programme constitué pour exécuter une fonction définie nécessaire à l'exécution d'une classe, plutôt que l'anglicisme *pack* j'ai retenu à la réflexion l'apocope *paq* qui me semble plus judicieuse. De plus l'homophonie - très rare - du terme avec son synonyme me comble...

---

RAYMOND ROCHEDIEU  
La Cornée du Der - avril 2017