Le package longtable *

David Carlisle†
traduction de Jean-Pierre Drucbert ‡

1998/05/13

Résumé

Ce package définit l'environnement longtable, une version « multi-page » de tabular.

Liste des tableaux

1 Introduction

`longtable` Ce package définit un nouvel environnement, longtable, qui possède la plupart des possibilités de l'environnement tabular, mais produit des tables pouvant dépasser la hauteur d'une page, c'est-à-dire qui peuvent être coupées par l'algorithme standard de rupture de page de TEX. Il partage aussi certaines possibilités de l'environnement table. En particulier, il utilise le même compteur, `table`, et a une commande `\caption` similaire. La commande `\listoftables` liste les tables produites par les environnements table et longtable.

L'exemple suivant utilise la plupart des caractéristiques de l'environnement longtable. Un listing source de cet exemple est donné dans la Section 8.

Note : Plusieurs parties de la table suivante *ne* seront *pas* alignées correctement jusqu'à ce que ce document soit compilé plusieurs fois par LATEX. C'est un trait caractéristique de ce package, comme décrit ci-dessous.

*Ce fichier a le numéro de version v4.09, révisé le 1998/05/13.

†Le nouvel algorithme pour aligner les morceaux (« chunks ») d'une table, utilisé dans la version 4 de ce package a été révisé, implémenté et documenté par David Kastrup, `dak@neuroinformatik.ruhr-uni-bochum.de`.

‡Traductions additionnelles : Françoise Marre-Fournier <marre-fournier@unilim.fr> ; relecture Benjamin Bayart <bayartb@edgard.fdn.fr>

Tab. 1: Une longue table

*	Cette partie apparaît au sommet de la table		*
*	Première	Seconde	*
*	Les colonnes de longtable sont spécifiées	de la	*
*	même manière que dans l'environnement	tabular.	*
*	`@{*}r||p{1in}@{*}`	dans ce cas.	*
*	Chaque ligne se termine par une	commande \\.	*
*	La commande \\ a un	argument	*
*	optionnel, tout comme dans l'environnement	tabular.	*
*	Voyez vous l'effet de `\\[10pt]`	?	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	De même `\hline` peut être utilisée,	comme dans tabular.	*
*	Ceci était une `\hline`	.	*
*	Ceci était une `\hline\hline`	.	*
‖	Ceci est une `\multicolumn{2}{||c||}`		‖
*	Si une rupture de page a lieu sur une `\hline` alors	une ligne est tracée	*
*	au bas d'une page et au	sommet de la suivante.	*
*	L'argument `[t] [b] [c]` de tabular	ne peut pas être utilisé.	*
*	L'argument optionnel doit être choisi parmi	`[l] [r] [c]`	*
*	pour spécifier si la table doit être	ajustée	*
*	sur la gauche, sur la droite	ou centrée.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Ceci va en	bas.	*

Tab. 1: (suite)

*	Cette partie apparaît en haut de chacune des autres pages		*
*	**Première**	**Seconde**	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Certaines lignes prennent beaucoup de place, comme celle-ci :	Cette dernière colonne est une colonne « p » si bien que cette « ligne » de la table peut occuper plusieurs lignes. Notez cependant que TEX ne coupera jamais une page à l'intérieur d'une telle ligne. Les ruptures de page ne se produisent qu'entre les rangées de la table ou sur des commandes `\hline`.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas[1] de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci[2]	*
*	Ceci va en	bas.	*

[1]Ceci est une note

[2]longtable prend des précautions spéciales et des notes de bas de page peuvent aussi être

Tab. 1: (suite)

*	Cette partie apparaît en haut de chacune des autres pages		*
*	**Première**	**Seconde**	*
*	Des tas de lignes	comme celle-ci.	*
*	Des tas de lignes	comme celle-ci.	*
*	Ces lignes	apparaîtront	*
*	à la place du	bas habituel	*
*	à la fin	de la table	*

2 Taille du pas ou « chunk size »

`LTchunksize`

Pour pouvoir traiter par TEX des tables faisant plusieurs pages, il est nécessaire de partager la table en plus petits morceaux (les « chunks », littéralement « bouchées », NdT), pour que TEX ne soit pas obligé de tout garder à la fois. Par défaut, longtable utilise 20 lignes par pas, mais l'usager peut choisir la valeur qu'il désire, avec par exemple, `\setcounter{LTchunksize}={10}`.[3] Ces pas ou « chunks » n'affectent pas la rupture des pages, donc si vous utilisez TEX avec beaucoup de mémoire, vous pouvez donner à `LTchunksize` l'équivalent de plusieurs pages de la table. TEX s'exécutera plus rapidement avec un grand `LTchunksize`. Cependant si cela est nécessaire longtable peut travailler avec `LTchunksize=1`, auquel cas la mémoire utilisée sera négligeable. Notez que si vous utilisez les commandes pour placer un en-tête en haut de la page (voir plus loin), la valeur de `LTchunksize` doit être au moins égale au nombre de lignes de l'en-tête ou du pied. (En pratique, une valeur entre 10 et 50 semble raisonnable. NdT).

Ce document spécifie `\setcounter{LTchunksize}{10}`. Si vous pouviez regarder la table précédente après la *première* exécution de LATEX vous verriez que les deux moitiés ne collent pas[4]. De plus LATEX a imprimé un avertissement comme quoi la largeur des colonnes a changé. longtable écrit des informations dans le fichier `.aux`, pour pouvoir aligner les différents pas ou « chunks ». Avant la version 4, il n'utilise ces informations que si une commande `\setlongtables` est donnée avant que la table commence. Cependant, maintenant, ces informations sont toujours utilisées, grace à un nouvel algorithme[5] et donc `\setlongtables` n'est plus utile. Il est défini (mais ne fait rien) pour que les anciens documents qui l'emploient puissent être utilisé.

utilisées (sans abus!) dans les colonnes 'p'.

[3] On peut aussi utiliser la syntaxe plain TEX `\LTchunksize=10`.

[4] En fait, vous ne voyez rien d'étrange ici, car au second passage de LATEX, tout est remis en place. Et ce document ne s'est pas fait en un seul passage (ni même en seul un jour). NdT

[5] Dû à David Kastrup

Un	environnement	tabular
dans	une table	flottante

TAB. 2 – Une table flottante

3 Légendes et en-têtes

Au début de la table vous pouvez spécifier des lignes qui devront apparaître en haut de chaque page (en dessous de la ligne de tête, mais avant les autres lignes de la table).

`\endhead` Ces lignes sont entrées normalement, mais la dernière commande `\\` est remplacée par une commande `\endhead`.

`\endfirsthead` Si la première page doit avoir une en-tête différente, alors celle-ci doit être donnée en premier, et terminée par la commande `\endfirsthead`. La valeur de `\LTchunksize` doit être choisie au moins égale au nombre de lignes de l'en-tête.
`\endfoot` `\endlastfoot` Il y a aussi des commandes `\endfoot` et `\endlastfoot` qui sont utilisées de la même façon (au *début* de la table) pour spécifier des lignes (ou une `\hline`) devant apparaître en bas de chaque page. Dans certaines situations, vous pourrez vouloir placer des lignes qui appartiennent logiquement au corps de la table à la fin de firsthead, ou au début de lastfoot. Ceci peut aider à contrôler quelles lignes apparaissent sur la première et la dernière page de la table.

`\caption` La commande `\caption{...}` est essentiellement équivalente à `\multicolumn{n}{c}{\parbox{\LTcapwidth}{...}}` où `n` est le nombre de colonnes de la table. Vous pouvez forcer la largeur de la légende avec une commande telle que `\setlength{LTcapwidth}{2in}` dans le préambule de votre document. La valeur par défaut est de quatre pouces. `\caption` écrit aussi les informations pour produire une entrée dans la liste des tables. Comme pour la commande `\caption` des environnements figure et table, un argument optionnel spécifie le texte devant apparaître dans la liste des tables si ce texte est différent de celui apparaissant dans la légende elle-même. Donc la légende pour la table 1 a été spécifiés par `\caption[Une légende optionnelle (pour la liste des tables)]{Une longue table\label{long}}` Vous pouvez désirer que la légende sur les pages suivantes soit différente de celle de la première page. Dans ce cas placez la commande `\caption` dans le premier en-tête, et une autre légende dans une commande `\caption[]` dans l'en-tête principal. Si l'argument optionnel de la commande `\caption` est vide, aucune entrée n'est ajoutée dans la liste des tables. De plus, si vous ne voulez pas que le numéro de table soit imprimé chaque fois, utilisez la commande `\caption*`.

Le légendes sont formattés suivant le code de la classe article. Si vous avez redéfini la commande standard `\@makecaption` pour produire un format différent de légende, vous serez peut-être amené à réaliser des changements similaires pour le `\LT@makecaption` de la version de longtable. Voir le code pour plus de détails. Une métode plus pratique pour personnaliser les légendes donnée par le package caption2, qui fournit des commandes de légendes personnalisés, et qui adapte les légendes dans les environnements standards. De nombreux autres environnements fournis par des packages (incluant le package longtable) sont modifiés de façon compatible.

Vous pouvez utiliser la commande `\label` pour faire des références croisés aux longues tables avec `\ref` (ou `\pageref`, NdT). Notez dependant que la commande `\label` ne doit pas être utilisée dans un en-tête qui peut apparaître plus d'une fois. Placez la, soit dans le fisrthead ou dans le corps de la table. Elle ne doit être la *première* commande de n'importe quelle entrée. [Les règles exactes sur les endroits où il est possible de placer la commande `\label` sont assez TEXniques, mais il est toujours possible et inoffensif de la placer *à l'intérieur* de l'argument d'une commande `\caption`. NdT].

4 Entrées multicolonnes

La commande `\multicolumn` peut être utlisée exactement de la même manière que dans tabular. Vous pouvez vouloir sauter cette section, qui est plutôt technique, cependant prendre en compte `\multicolumn` est un des principaux problèmes pour un environnement tel que longtable. Le principal effet que l'utilisateur verra est que certaines combinaisons des entrées `\multicolumn` auront comme résultat que le document devra être plusieurs fois compilé par LATEX avant que les différents morceaux (« chunks ») d'une table ne soient alignés.

Les exemples de cette section sont réalisés en utilisant un `LTchunksize` fixé à la valeur minimum, pour montrer les effects lorsque les entrés `\multicolumn` se produisent dans différents morceaux.

Considérons la Table 3. Dans le deuxième morceau, longtable voit l'entré large de la multicolonne. À ce point, il considère que les deux premières colonnes sont trés étroites. Toute la largeur de l'entrée multicolonne est supposée être dans la troisième colonne (c'est une « caractéristique » de la commande originelle `\halign` de TEX). Ensuite longtable prend en compte l'information selon laquelle il y a une troisième colonne large pour le dernier morceau. Le résultat est que la première « passe » est trop « large ». Si les « lignes sauvées » à partir de cette première passe sont réinsérées dans la table à la passe suivante, la table sera alignée en deux passes mais pourra être beaucoup trop large.

`\kill` La solution de ce problème, employée dans les versions 1 et 2, est d'utiliser une « `\kill` line » (littéralement, une ligne tuée, NdT). Si une ligne est tuée (`\kill`ed), par l'utilisation de `\kill`, au lieu de `\\`, à la fin de la ligne, cela est utilisé dans le calcul des largeurs de colonne, mais enlevé de la table finale. Donc, l'entrée de copies tuées (`\kill`ed copies) des deux dernières lignes, avant l'entrée entière de la colonne, devrait signifier que `\halign` « voyait » l'entrée complète dans les deux premières colonnes, et donc, élargissait la troisième colonne de façon à faire la place à l'entrée multicolonne.

Dans la version 3, une nouvelle solution a été introduite. Si la ligne sauvée dans le fichier `.aux` n'a pas été utilisée, longtable utilisait une forme « brouillon » de `\multicolumn` ; cela modifiait la définition, de façon à ce que l'entrée étendue ne soit pas plus large que la colonne élargie. Donc, après la première passe, le fichier `.aux` stockait la plus large entrée normale pour chaque colonne, aucune colonne

Tab. 3: Une combinaison difficile de `\multicolumn` : pass 1

1	2	3
une multicolonne trés large 1–3		
une multicolonne 1–2		3
wide 1	2	3

Tab. 4: Une combinaison difficile de `\multicolumn` : pass 2

1	2	3
une multicolonne trés large 1–3		
une multicolonne 1–2		3
wide 1	2	3

Tab. 5: Une combinaison difficile de `\multicolumn` : pass 3

1	2	3
une multicolonne trés large 1–3		
une multicolonne 1–2		3
wide 1	2	3

Tab. 6: Une combinaison difficile de `\multicolumn` : pass 4

1	2	3
une multicolonne trés large 1–3		
une multicolonne 1–2		3
wide 1	2	3

n'étant élargie à cause des colonnes `\span`ées. Par défaut, longtable ignorait le fichier `.aux`, et donc chaque compilation par LaTeX était considérée comme la première passe. Une fois la déclaration de `\setlongtables` faite, la ligne sauvée dans le fichier `.aux` et la définition correcte de `\multicolumn` était utilisées. Si une entrée `\multicolumn` entraînait l'élargissement d'une colonne, cette information ne pouvait être ignorée pour le morceau suivant, et donc la table ne pouvait pas être correctement alignée jusqu'à la troisième passe. Cette algorithme convergeait toujours en trois passes, comme décrit ci-dessus, mais dans les exemples tels qu'un de ceux des tables 3–6, les largeurs finales n'étaient pas optimales pour celle de la colonne 2 : en effet, cette largeur est déterminée par une entrée `\multicolumn` qui n'est pas connue quand la largeur finale de la colonne 3 a été déterminée car les *deux* commandes `\multicolumn` sont interverties entre les modes « brouillon » et « normal » en même temps.

La version 4 allège considérablement le problème. La première compilation de la table aura en effet une troisième colonne beaucoup trop large. Cependant, à la passe suivante, longtable prendra en compte l'erreur et réduira la largeur de colonne de façon conforme. Si cela doit déplacer le morceau avant celui de `\multicolumn`, une passe additionnelle, bien sûr, sera nécessaire.

Il est possible de construire des tables dans lesquelles cette détermination en cascade des bonnes largeurs nécessite plusieurs compilation pour « converger » et produire une table avec tous ses morceaux alignés. Cependant, pour avoir besoin de plusieurs compilations, il faut construire une table avec de nombreuses entrées `\multicolumn` se chevauchant, toutes étant plus larges que la largeur naturelle des colonnes à étendre et toutes se produisant dans des parties de table différentes.

Dans le cas le plus courant, l'algorithme convergera après deux ou trois passes ; on aura l'avantage de ne pas rééditer le document avant la compilation finale pour ajouter `\setlongtables` et le meilleur choix de la largeur finale des colonnes, en cas de plusieurs entrées `\multicolumn` sera, heureusement, plus aisé que les passes supplémentaires qui seraient nécessaires.

Donc, la table 3 converge après 4 passes comme on peut le voir table 6.

Vous pouvez encore accélérer la convergence en introduisant judicieusement des lignes `\kill`ées, s'il vous arrive d'avoir des tables morcellées, comme ci-dessus.

Et si vous ne désirez pas compiler un fichier deux fois, vous devez, à la première ligne de chaque longtable, saisir une ligne `\kill`ée qui contient l'entrée la plus large qui sera utilisée dans chaque colonne. Tous les morceaux seront alignés à la première compilation.

5 Ajustement

`\LTleft` `\LTright`

L'argument optionnel de longtable contrôle l'alignement horizontal de la table. Les options possibles sont `[c]`, `[r]` et `[l]`, pour centrer, aligner à droite et aligner à gauche, respectivement. Normalement, le centrage est l'option par défaut, mais ce document spécifie

```
\setlenght\LTleft\parindent
\setlenght\LTright\fill
```

dans son préambule, ce qui signifie que les tables sont cadrées à gauche, mais indentées comme pour un paragraphe normal. Des longueurs quelconques peuvent être spécifiées pour ces paramètres, mais au moins l'une d'elles devrait être élastique pour pouvoir remplir la largeur de la page, à moins que des longueurs élastiques soient ajoutées entre les colonnes en utilisant la commande `\extracolumnsep`. Par exemple

```
\begin{tabular*}
{\textwidth}{@{\extracolsep{...}}...}
```

produit un table de pleine largeur, pour produire un effet similaire avec `longtable` :

```
\setlength\LTleft{0pt}
\setlength\LTright{0pt}
\begin{longtable}{@{\extracolsep{...}}...}
```

6 Changements

Cette section présente les principaux changements depuis la version 2. Un état plus détaillé des changements peut être produit à la fin du listing source si le fichier `ltxdoc.cfg` spécifie

```
\AtBeginDocument{\RecordChanges}
\AtEndDocument{\PrintChanges}
```

Changements effectués entre les versions 2 et 3 :

- Le mécanisme d'ajout des en-tête et pied de table a été complètement réécrit. Avec ce nouveau code, longtable ne nécessite pas de `\clearpage` au début de la page, ainsi une table peut commencer au milieu d'une page. De plus, la commande `\endlastfoot`, qui n'était pas implémentée de façon solide dans l'ancien code, a été ajoutée ;
- longtable provoque maintenant une erreur s'il démarre dans un contexte `\twocolumn` ou dans l'environnement multicols ;
- le fichier séparé de documentation `longtable.tex` a été fusionné avec le fichier d'extension, `longtable.dtx` en utilisant le package doc de Mittelbach ;
- la prise en compte des notes de bas de page a été ajoutée. Cependant, il est à noter que `\footnote` ne fonctionnera pas dans les sections d'« en-tête » ou de « pied » de table. Pour pouvoir utiliser une note de bas de page dans ces sections (par exemple, dans une légende), il faut utiliser `\footnotemark` au point d'insertion et `\footnotetext`, n'importe où dans le *corps* de la table, qui apparaîtra sur la même page ;
- le traitement de `\multicolumn` a été modifié, rendant inutile de « tuer » (`\kill`) les lignes, au prix, quelquefois, de réaliser une troisiéme compilation par LaTeX ;
- la commande `\newpage` fonctionne maintenant dans l'environnement longtable.

Changements effectués entre les versions 3 et 4 :

- Un nouvel algorithme est utilisé pour aligner les morceaux. En plus de la plus grande largeur de chaque colonne, longtable se souvient de quel morceau produisait ce maximum. Cela lui permet de vérifier que le maximum est encore atteint dans les derniéres compilations. Comme longtable peut maintenant traiter les colonnes retrécies pendant que le fichier est édité, la commande `\setlongtables` n'est plus nécessaire et est désactivée.
- Un bénéfice supplémentaire de la nouvelle capacité de l'algorithme à traiter les colonnes rétrécies est qu'il produit de meilleures (par rapport l'algorithme précédent) largeurs de colonne (plus étroites) dans le cas d'entrées de `\multicolumn` se chevauchant dans différents morceaux ;
- le mode « brouillon » de `\multicolumn` a été enlevé suivant les commandes asociées telles que `\LTmulticolumn` ;
- l'inconvénient du nouvel algorithme est qu'il peut faire plus de passes. Le maximum théorique est approximativement deux fois la longueur d'un « enchaînement » de colonnes avec des entrées `\multicolumn` se chevauchant, bien qu'en pratique il converge habituellement plus vite que l'ancienne version (qui converge toujours en trois passes une fois la commande `\setlongtables` activée) ;
- les commandes `\\*` et `\nopagebreak` peuvent être utilisées pour contrôler les sauts de page.

7 Résumé

TAB. 7: Résumé des commandes longtable

	Paramètres	
`\LTleft`	Glu sur la gauche de la table.	(`\fill`)
`\LTright`	Glu sur la droite de la table.	(`\fill`)
`\LTpre`	Glu avant la table.	(`\bigskipamount`)
`\LTpost`	Glu après la table.	(`\bigskipamount`)
`\LTcapwidth`	La largeur de la « parbox » contenant la légende.	(4in)
`LTchunksize`	Le nombre de ligne par pas ou « chunk ».	(20)
	Arguments optionnels de `\begin{longtable}`	
aucun	Position spécifiés par `\LTleft` et `\LTright`.	
`[c]`	Centre la table.	
`[l]`	Place la table sur la gauche.	
`[r]`	Place la table sur la droite.	
	Commandes à la fin des lignes de la table	
`\\`	Spécifie la fin d'une ligne	
`\\[`⟨*dim*⟩`]`	Finit la ligne, puis ajoute un espace vertical (comme dans l'environnement tabular).	
`\\*`	Identique à `\\` mais interdit un saut de page après la ligne.	

`\tabularnewline`	Équivalent à `\\`. À utiliser dans un contexte `\raggedright` (ou similaire) qui redéfinit `\\`.
`\kill`	La ligne est effacée mais est utilisée dans le calcul des largeurs.
`\endhead`	Spécifie les lignes qui doivent apparaître en haut de chaque page.
`\endfirsthead`	Spécifie les lignes qui doivent apparaître en haut de la première page.
`\endfoot`	Spécifie les lignes qui doivent apparaître en bas de chaque page.
`\endlastfoot`	Spécifie les lignes qui doivent apparaître en bas de la dernière page.

Les commandes de légende de **longtable**

`\caption{`⟨*caption*⟩`}`	Légende 'Table ? : ⟨*caption*⟩', et une entrée '⟨*caption*⟩' dans la liste des tables.
`\caption[`⟨*lot*⟩`]{`⟨*caption*⟩`}`	Légende 'Table ? : ⟨*caption*⟩', et une entrée '⟨*lot*⟩' dans la liste des tables.
`\caption[]{`⟨*caption*⟩`}`	Légende 'Table ? : ⟨*caption*⟩', mais pas d'entrée dans la liste des tables.
`\caption*{`⟨*caption*⟩`}`	Légende '⟨*caption*⟩', mais pas d'entrée dans la liste des tables.

Commandes disponibles au début d'une ligne

`\pagebreak`	Force un saut de page.
`\pagebreak[`⟨*val*⟩`]`	Un poids (« hint ») entre 0 et 4 pour autoriser un saut.
`\nopagebreak`	Interdit un saut de page .
`\nopagebreak[`⟨*val*⟩`]`	Un poids (« hint ») entre 0 et 4 pour interdire un saut.
`\newpage`	Force un saut de page.

Commandes de note de bas de page disponible dans longtable

`\footnote`	Note de bas de page, mais qui ne doit pa être utilisée dans les têtes et pieds de tables.
`\footnotemark`	Marque de la note de bas de page, peut être utilisée dans les têtes et pieds de tables.
`\footnotetext`	Texte de la note de bas de page, à utiliser dans le corps de la page.

Setlongtables

`\setlongtables`	Commande obsolète. Ne fait plus rien.

8 Source de la Table 1

```
\begin{longtable}{@{*}r||p{1in}@{*}}
KILLED & LINE!!!! \kill
\caption
[Un caption optionnel (pour la liste des tables)]
{Une longue table\label{long}}\\
\hline\hline
\multicolumn{2}{@{*}c@{*}}%
{Cette partie appara\^{\i}t au sommet de la table}}\\
\textsc{Premi\`ere}&\textsc{Seconde}\\
\hline\hline
\endfirsthead
\caption[]{(suite)}\\
\hline\hline
\multicolumn{2}{@{*}c@{*}}%
      {Cette partie appara\^{\i}t en haut de chacune des autres pages}\\
\textbf{Premi\`ere}&\textbf{Seconde}\\
\hline\hline
\endhead
\hline
Ceci va en&bas.\\
\hline
\endfoot
\hline
Ces lignes&appara\^{\i}tront\\
\`a la place du&bas habituel\\
\`a la fin&de la table\\
\hline
\endlastfoot
Les colonnes de \env{longtable} sont sp\'ecifi\'ees& de la\\
m\^eme mani\`ere que dans l'environnement& \env{tabular}.\\
...
\multicolumn{2}{||c||}%
{Ceci est une ...}\\
...
  Certaines lignes prennent beaucoup de place, comme celle-ci: &
   \raggedleft Cette derni\`ere colonne est une colonne <<~p~>> \\
   si bien que cette <<~ligne~>> de la table peut occuper \\
   plusieurs lignes. Notez cependant que \TeX\ ne coupera \\
   jamais une page \`a l'int\'erieur d'une telle ligne. \\
   Les ruptures de page ne se produisent qu'entre les rang\'ees \\
   de la table ou sur des commandes "\hline".\\
   \tabularnewline
...
Des tas de lignes& comme celle-ci.\\
...
\hline
Des tas\footnote{...} de lignes& comme celle-ci.\\
Des tas de lignes& comme celle-ci\footnote{...}\\
```

```
\hline
Des tas de lignes& comme celle-ci.\\
...
\end{longtable}
```

9 Les macrocommandes

```
⟨*package⟩
```

9.1 Code initial

Avant de déclarer les options du package, nous devons définir les valeurs par défaut ici.

`\LT@err` La commande de génération d'erreur

```
\def\LT@err{\PackageError{longtable}}
```

`\LT@warn` La commande de génération des alertes

```
\def\LT@warn{\PackageWarning{longtable}}
```

`\LT@final@warn` Si des longtables ne sont pas alignées, génère une alerte à la fin de la compilation, `\AtEndDocument`.

```
\def\LT@final@warn{%
  \AtEndDocument{%
    \LT@warn{Table \@width s have changed. Rerun LaTeX.\@gobbletwo}}%
  \global\let\LT@final@warn\relax}
```

9.2 Options

Les deux premir̀es options traitent le maniement des erreurs. Elles sont compatibles avec les options du package `tracefnt`

`errorshow` Affiche *seulement* les erreurs à l'écran. Les messages d'alertes apparaissent uniquement dans le fichier log.

```
\DeclareOption{errorshow}{%
  \def\LT@warn{\PackageInfo{longtable}}}
```

`pausing` Dans une erreur, tous les messages d'alerte stoppent la compilation par TEX. Peut être utile pour le debuggage.

```
\DeclareOption{pausing}{%
  \def\LT@warn#1{%
    \LT@err{#1}{This is not really an error}}}
```

`set`
`final` Les options suivantes présentent seulement une syntaxe alternative pour la déclaration de `\setlongtables`.

```
\DeclareOption{set}{}
\DeclareOption{final}{}
```

```
\ProcessOptions
```

9.3 Paramètres utilisateur

`\LTleft`
`\LTright` Glu sur la gauche ou la droite de la table, `\fill` par défaut (*i.e.* centrée).

```
\newskip\LTleft        \LTleft=\fill
\newskip\LTright       \LTright=\fill
```

`\LTpre` `\LTpost` Glu avant et après longtable. Par défaut, `\bigskip`.

```
\newskip\LTpre        \LTpre=\bigskipamount
\newskip\LTpost       \LTpost=\bigskipamount
```

`\LTchunksize` Taille du pas (« chunk ») : le nombre de ligne donné par `\halign`, 20 par défaut.

```
\newcount\LTchunksize \LTchunksize=20
```

`\c@LTchunksize` Ajouté dans la V3.07 pour autoriser la syntaxe LaTeX `\setcounter{LTchunksize}{10}`.

```
\let\c@LTchunksize\LTchunksize
```

`\LTcapwidth` Largeur de la `\parbox` contenant la légende, 4in par défaut.

```
\newdimen\LTcapwidth  \LTcapwidth=4in
```

9.4 Paramètres internes

`\LT@head` `\LT@firsthead` `\LT@foot` `\LT@lastfoot` Boîtes pour les tête et pied de table.

```
\newbox\LT@head
\newbox\LT@firsthead
\newbox\LT@foot
\newbox\LT@lastfoot
```

`\LT@cols` Compteur de numéro des colonnes.

```
\newcount\LT@cols
```

`\LT@rows` Compteur du pas (« chunksize »)

```
\newcount\LT@rows
```

`\c@LT@tables` Compteur de table, ajouté dans la V3.02. Les versions précédentes utilisaient seulement de compteur `table` de LaTeX mais cela échouait si `table` était remis à zéro pendant la compilation ; par exemple, la classe report le remet à zéro dans chaque chapitre.

Cela a été modifié, dans la V3.04, par l'utilisation de `\newcount\LT@tables`. Les compteurs LaTeX sont correctement préservés lorsque `\includeonly` est utilisé. Dans le reste du fichier, `\LT@tables` a été remplacé par `\c@LT@tables` sans autre commentaire supplémentaire.

```
\newcounter{LT@tables}
```

`\c@LT@chunks` Nous avons besoin de compter les pas d'une table depuis la Version 4.

```
\newcounter{LT@chunks}[LT@tables]
```

`\c@table` `\fnum@table` `\tablename` Si le compteur `table` n'est pas défini (par exemple dans le style `letter`), cette macro le définit (Ajouté dans la V3.06).

```
\ifx\c@table\undefined
  \newcounter{table}
  \def\fnum@table{\tablename~\thetable}
\fi
\ifx\tablename\undefined
  \def\tablename{Table}
\fi
```

\LT@out Dans les styles standards, `longtable` utilise le fichier `.aux` pour enregistrer les largeurs de colonne. Avec `letter.sty`, le package utilise un fichier séparé `.lta`. (Ajouté dans la V3.06).

Inutile pour la nouvelle classe `letter`.

```
\ifx\startlabels\undefined
  \let\@auxout\@auxout
\else
  {\@input{\jobname.lta}}%
  \newwrite\@auxout
  \immediate\openout\@auxout=\jobname.lta
\fi
```

\LT@p@ftn Stockage temporaire du texte de la note de bas de page dans un colonne « p ».

```
\newtoks\LT@p@ftn
```

\LT@end@pen Poids (« penalty ») spécial pour la fin de la table. Réalisé comme ça pour épargner un compteur.

```
\mathchardef\LT@end@pen=30000
```

9.5 L'environnement longtable

\longtable Appelé par `\begin{longtable}`. Cette implémentation ne fonctionne pas dans un format à colonnes multiples. `\par` ajouté dans la V3.04.

```
\def\longtable{%
  \par
  \ifx\multicols\@undefined
  \else
     \ifnum\col@number>\@ne
       \@twocolumntrue
     \fi
  \fi
  \if@twocolumn
    \LT@err{longtable not in 1-column mode}\@ehc
  \fi
  \begingroup
```

Vérifie la présence d'un argument optionnel.

```
  \@ifnextchar[\LT@array{\LT@array[x]}}
```

\LT@array Commence l'ajustement de l'alignement. Basé sur `\@array` du noyau LaTeX et sur le package array

Depuis la Version 3.02, longtable utilise le compteur interne `\c@LT@tables`. Le compteur `table` de LaTeX est encore incrémenté pour que `\caption` fonctionne correctement.

```
\def\LT@array[#1]#2{%
  \refstepcounter{table}\stepcounter{LT@tables}%
```

Crée la justification (« glu ») autour de la table, si un argument optionnel est donné.

```
\if l#1%
  \LTleft\z@ \LTright\fill
\else\if r#1%
  \LTleft\fill \LTright\z@
\else\if c#1%
  \LTleft\fill \LTright\fill
\fi\fi\fi
```

Fixe ces commandes internes pour longtable.

```
\global\let\LT@mcw@rn\relax
```

```
\let\LT@mcol\multicolumn
```

Maintenant redéfinit `\@tabarray`, pour rétablir `\hline` et `\multicolumn`, pour que array et tabular encapsulés dans longtable (ou dans l'en-tête de page d'un longtable) donnent le résultat escompté. Les définitions originales sont ici sauvegardées pour que vous puissiez charger le package array avant ou après longtable.

```
\let\LT@@tabarray\@tabarray
\let\LT@@hl\hline
\def\@tabarray{%
  \let\hline\LT@@hl
```

```
\let\multicolumn\LT@mcol
```

```
  \LT@@tabarray}%
\let\\\LT@tabularcr\let\tabularnewline\\%
\def\newpage{\noalign{\break}}%
```

Définitions plus ou moins standards, mais commencent d'abord un `\noalign`.

```
\def\pagebreak{\noalign{\ifnum`}=0\fi\@testopt{\LT@no@pgbk-}4}%
\def\nopagebreak{\noalign{\ifnum`}=0\fi\@testopt\LT@no@pgbk4}%
```

```
\let\hline\LT@hline \let\kill\LT@kill\let\caption\LT@caption
\@tempdima\ht\strutbox
```

```
\let\@endpbox\LT@endpbox
```

Commandes internes établies en accord avec celles de Lamport ou celles de Mittelbach.

```
\ifx\extrarowheight\@undefined
```

Initialise ces commandes comme dans tabular du noyau LaTeX.

```
  \let\@acol\@tabacol
  \let\@classz\@tabclassz \let\@classiv\@tabclassiv
  \def\@startpbox{\vtop\LT@startpbox}%
  \let\@@startpbox\@startpbox
  \let\@@endpbox\@endpbox
  \let\LT@LL@FM@cr\@tabularcr
\else
```

Initialise ces commandes comme dans array. `\d@llar`remplacé par `\d@llarbegin` `\d@llarend` dans la V3.03 pour correspondre à array V2.0h. Nous n'avons pas besoin de fixer la valeur de `\d@llarbegin` et `\d@llarend` puisque le package array leur donne leurs valeurs correctes au premier niveau.

```
  \advance\@tempdima\extrarowheight
  \col@sep\tabcolsep
  \let\@startpbox\LT@startpbox\let\LT@LL@FM@cr\@arraycr
\fi
```

La reste de cette macro est principalement basé sur le package array, mais devrait aussi fonctionner pour l'environnement tabular standard.

```
\setbox\@arstrutbox\hbox{\vrule
  \@height \arraystretch \@tempdima
  \@depth \arraystretch \dp \strutbox
  \@width \z@}%
\let\@sharp##\let\protect\relax
```

Interpréte l'argument du préambule.

```
 \begingroup
  \@mkpream{#2}%
```

Nous avons besoin de renommer ici `\@preamble` puisque le système de F. M. utilise `\global`, et nous avons peut être besoin d'encapsuler `\@mkpream`, par exemple pour `\multicolumn` ou pour un array. Nous n'avons pas besoin de nous préoccuper des longtables encapsulées.

```
  \xdef\LT@bchunk{%
```

```
     \global\advance\c@LT@chunks\@ne
     \global\LT@rows\z@\setbox\z@\vbox\bgroup
```

La ligne suivante a été ajoutée dans la v4.05. Pour que `\penalties` soit pris en compte aux limites des morceaux, il est nécessaire de se soucier de où et quand `\lineskip` est ajouté. La commande suivante ne fait rien en haut de la table, ni dans les pas d'en-tête, mais dans les morceaux du corps de la table, cela fixe `\prevdepth` (à 0pt, mais n'importe quelle valeur convient) pour que `\lineskip` soit ajoutée. La chose importante à noter est que la glu sera ajoutée *après* n'importe quelle donnée verticale provenant de `\noalign`.

```
     \LT@setprevdepth
```

```
     \tabskip\LTleft\halign to\hsize\bgroup
    \tabskip\z@ \@arstrut \@preamble \tabskip\LTright \cr}%
\endgroup
```

Découvre combien nous avons de colonnes (stocké dans `\LT@cols`).

```
\expandafter\LT@nofcols\LT@bchunk&\LT@nofcols
```

Obtient les lignes sauvées à partir de `\LT@i`...`\LT@ix` (à partir du fichier `.aux`), ou insère une nouvelle ligne vide.

```
\LT@make@row
```

Quelques commandes internes supplémentaires pour longtable.

```
\m@th\let\par\@empty
\everycr{}\lineskip\z@\baselineskip\z@
```

Débute le premier morceau.

```
  \LT@bchunk}
```

`\LT@no@pgbk` Peut simplifier le `\@no@pgbk` standard puisqu'il est seulement en vmode, mais nécessite alors de terminer `\noalign`.

```
\def\LT@no@pgbk#1[#2]{\penalty #1\@getpen{#2}\ifnum`{=0\fi}}
```

`\LT@start` Cette macro débute le procédé de placement de la table dans la page courante. Il n'est pas appelé jusqu'à ce qu'une commande `\\` ou `\endlongtable` termine un morceau, puisque nous ne connaissons pas encore laquelle des quatre sections possibles d'en-tête ou de pied a été définie.

Elle commence par se redéfinir elle-même, de façon à ce que la table soit commencée une seule fois ! Dans la V3.04, elle était redéfinie en `\relax`, maintenant, elle utilise `\endgraf` pour forcer l'activation de saut de page.

```
\def\LT@start{%
  \let\LT@start\endgraf
  \endgraf\penalty\z@\vskip\LTpre
```

Commence une nouvelle page s'il n'y a pas assez de place pour l'en-tête et le pied de page et pour une ligne supplémentaire.

```
  \dimen@\pagetotal
  \advance\dimen@ \ht\ifvoid\LT@firsthead\LT@head\else\LT@firsthead\fi
  \advance\dimen@ \dp\ifvoid\LT@firsthead\LT@head\else\LT@firsthead\fi
  \advance\dimen@ \ht\LT@foot
```

À ce point, j'avais l'habitude d'ajouter `\ht\@arstrutbox` et `\dp\@arstrutbox` pour mesurer la taille d'une ligne. Cependant, cela peut échouer de façon spectaculaire pour une colonne `p` qui peut être beaucoup plus large. Les versions précédentes pouvaient finir par créer une table commençant par un pied de table, puis un saut de page puis un en-tête puis, *enfin* le « premier en-tête » ! Donc, maintenant, on mesure la première ligne de la table, précisément, en la séparant verticalement (`\vsplit`ting) du premier morceau.

```
  \dimen@ii\vfuzz
  \vfuzz\maxdimen
    \setbox\tw@\copy\z@
    \setbox\tw@\vsplit\tw@ to \ht\@arstrutbox
    \setbox\tw@\vbox{\unvbox\tw@}%
  \vfuzz\dimen@ii
  \advance\dimen@ \ht
        \ifdim\ht\@arstrutbox>\ht\tw@\@arstrutbox\else\tw@\fi
  \advance\dimen@\dp
        \ifdim\dp\@arstrutbox>\dp\tw@\@arstrutbox\else\tw@\fi

  \advance\dimen@ -\pagegoal
  \ifdim \dimen@>\z@\vfil\break\fi
```

Stockage de la hauteur de la page moins le pied de table dans `\@colroom`.

```
      \global\@colroom\@colht
```

Si le pied n'est pas vide, réduction de façon homogène de `\vsize` et `\@colroom`.

```
\ifvoid\LT@foot\else
  \advance\vsize-\ht\LT@foot
  \global\advance\@colroom-\ht\LT@foot
  \dimen@\pagegoal\advance\dimen@-\ht\LT@foot\pagegoal\dimen@
  \maxdepth\z@
\fi
```

Place l'en-tête de table sur la page puis passe à le nouvelle routine de sortie.

```
\ifvoid\LT@firsthead\copy\LT@head\else\box\LT@firsthead\fi
\output{\LT@output}}
```

`\endlongtable` Appelé par `\end{longtable}`.

```
\def\endlongtable{%
```

Ajoute essentiellement un `\\` final. Mais comme nous connaissons le nombre réel de morceaux, nous pouvons enlever toutes les entrées se référant à une entrée maximale hors de la table (cela se produit seulement si la table a été raccourcie ou que la numérotation de table a disparu). Dans ce cas, nous devons, au moins, commencer par récupérer la nouvelle information correcte pour le dernier morceau de cette table, en enlevant la contrainte de largeur.

```
\crcr
\noalign{%
  \let\LT@entry\LT@entry@chop
  \xdef\LT@save@row{\LT@save@row}}%
\LT@echunk
\LT@start
\unvbox\z@
\LT@get@widths
```

Écrit la colonne modèle dans le fichier `.aux`. Depuis la V3.06, utilisation de fichier `.lta` pour `letter.sty`.

```
\if@filesw
  {\let\LT@entry\LT@entry@write\immediate\write\@auxout{%
```

Depuis la Version 3.02, longtable utilise le compteur interne `\c@LT@tables` plutôt que le compteur table de LaTeX. Cette information semble totalement différente de celle de la version 3. Cependant, nous n'avons pas besoin de renommer la macro car le code le plus récent considèrera qu'il n'y a pas de colonne et donc effacera les anciennes données.

```
    \gdef\expandafter\noexpand
      \csname LT@\romannumeral\c@LT@tables\endcsname
        {\LT@save@row}}}%
\fi
```

À ce point, émission d'un message d'alerte si une `\multicolumn` a été fixée en mode brouillon.

```
\LT@mcw@rn
```

Si le dernier morceau a des largeurs différentes par rapport au premier, alerte de l'utilisateur. De plus déclenchement d'une alerte pour recompiler avec LaTeX à la fin du document.

```
\ifx\LT@save@row\LT@@save@row
\else
  \LT@warn{Column \@width s have changed\MessageBreak
           in table \thetable}%
  \LT@final@warn
\fi
```

Force one more go with the longtable output routine.

```
\endgraf\penalty -\LT@end@pen
```

Maintenant, oblige le groupe à revenir à une routine standard.

```
\endgroup
```

Remise à zéro de `\@mparbottom` pour que les marges de paragraphes soient proches de la fin de la table.[6]

```
\global\@mparbottom\z@
\pagegoal\vsize
\endgraf\penalty\z@\addvspace\LTpost
```

Note de bas de page, comme cela est réalisé dans le package multicol.

```
\ifvoid\footins\else\insert\footins{}\fi}
```

9.6 Comptage des colonnes

Les colonnes sont comptées par l'inspection de `\@preamble`, plutôt que, simplement, par l'obtention de `\@mkpream`, pour incrémenter le compteur. De cette façon, le préambule est construit pour que le package soit compatible avec la plupart de ceux qui ajoutent une définition supplémentaire de colonne au standard LaTeX.

La version 1 comptait les `\@sharp` pour calculer le nombre de colonnes, cela a été changé dans la version 2 puisque cela ne fonctionnait pas avec NFSS. Maintenant, comptage des `&` [`lfonts.new` (et maintenant la définition selon le Standard LaTeX) définit `\@tabclassz` pour que `\@sharp` soit à l'intèrieur d'un groupe.]

`\LT@nofcols` Trouve le `&` suivant, puis regarde plus avant pour voir quel est le suivant.

```
\def\LT@nofcols#1&{%
  \futurelet\@let@token\LT@n@fcols}
```

`\LT@n@fcols` Ajoute un, puis stoppe au niveau d'un `\LT@nofcols` ou cherche le `&` suivant. L'astuce `\expandafter` a été ajoutée et le nom de `\@LT@nofcols` a été modifié pour conserver la convention d'appellation, `\LT@`.

```
\def\LT@n@fcols{%
  \advance\LT@cols\@ne
  \ifx\@let@token\LT@nofcols
    \expandafter\@gobble
  \else
    \expandafter\LT@nofcols
  \fi}
```

[6] Cela peut ne pas être correct. Cependant, si c'est oublié, il y a un problème avec les marges de paragraphe, [par exemple, page 3 de la documentation en V. O., NdT]. Y a-t'il des gourous des routine de sortie dans la salle ?

9.7 Les commandes \\ et \kill

\LT@tabularcr La définition interne de \\. Dans la forme *, insère un \nobreak après le \cr (ou \crcr) suivant.

Le traitement de la forme « étoile » a finalement été ajouté dans la v4.05. Durant les six ou sept dernières années, le commentaire de ce point disait

> Cette définition accepte aussi * qui agit de le même façon que \\. tabular fait cela mais longtable devra probablement définir * pour prévenir les sauts de page.

{\ifnum0=`}\fi ajouté dans la version 3.01, requis si la première entrée est vide. En fait, c'est insuffisant, puisque avec le package array, un groupe {} peut être introduit en mode mathématique. Donc, utilise la modification suivante. Ajoutée dans la v3.14

```
\def\LT@tabularcr{%
  \relax\iffalse{\fi\ifnum0=`}\fi
  \@ifstar
    {\def\crcr{\LT@crcr\noalign{\nobreak}}\let\cr\crcr
     \LT@t@bularcr}%
    {\LT@t@bularcr}}
```

\LT@crcr

```
\let\LT@crcr\crcr
```

\LT@setprevdepth Cela sera redéfini pour fixer \prevdepth au début d'un morceau.

```
\let\LT@setprevdepth\relax
```

\LT@t@bularcr

```
\def\LT@t@bularcr{%
```

Incrémente le compteur et insère le \\ de tabular ou termine le morceau. L'astuce \expandafter a été ajouté dans la version 3. Fixe \prevdepth au début d'un nouveau morceau. (C'est fait ici et pas dans le morceau en tête).

```
  \global\advance\LT@rows\@ne
  \ifnum\LT@rows=\LTchunksize
    \gdef\LT@setprevdepth{%
      \prevdepth\z@\global
      \global\let\LT@setprevdepth\relax}%
    \expandafter\LT@xtabularcr
  \else
    \ifnum0=`{}\fi
    \expandafter\LT@LL@FM@cr
  \fi}
```

\LT@xtabularcr Cela cherche juste un argument optionnel.

```
\def\LT@xtabularcr{%
  \@ifnextchar[\LT@argtabularcr\LT@ntabularcr}
```

\LT@ntabularcr La version sans argument optionnel. Ajoutée dans la version 3.01. Modifiée dans la v3.14.

```
\def\LT@ntabularcr{%
  \ifnum0=`{}\fi
  \LT@echunk
  \LT@start
  \unvbox\z@
  \LT@get@widths
  \LT@bchunk}
```

`\LT@argtabularcr` La version avec un argument optionnel. `\ifnum0=`{\fi}` dans la version 3.01. Modifié dans la v3.14.

```
\def\LT@argtabularcr[#1]{%
  \ifnum0=`{}\fi
  \ifdim #1>\z@
    \unskip\@xargarraycr{#1}%
  \else
    \@yargarraycr{#1}%
  \fi
```

Ajoute la ligne modèle et termine `\halign`.

```
  \LT@echunk
  \LT@start
  \unvbox\z@
  \LT@get@widths
  \LT@bchunk}
```

`\LT@echunk` Cela finit le morceau en cours et efface la ligne modèle.

```
\def\LT@echunk{%
  \crcr\LT@save@row\cr\egroup
  \global\setbox\@ne\lastbox
```

La ligne suivante a été ajoutée dans la v4.05. longtable dépend du paramètre `\lineskip` (qui est 0pt) pour fournir des points de saut entre chaque ligne pour que la table puisse être divisée en plusieurs pages.

Les versions précédentes laissaient le paramètre `\lineskip` à la fin de chaque morceau ajouté quand la ligne modèle est ajoutée. Il n'y avait pas de glu au début du morceau suivant puisque normalement TEX ne met pas de glu `\lineskip` en haut d'une boîte. Cela signifiait que les morceaux étaient parfaitement ajustés, cependant `\noalign` à la limite d'un morceau venait avant la première ligne du morceau suivant ou après l'interligne à la fin de ce morceau. C'est une mauvaise place, par exemple, cela signifie même que `\penalty10000` n'interdit pas un saut puisque la glu `\lineskip`, dans l'item précédent de la liste, donne un point de saut autorisé. Donc, maintenant, on enlève la glu `\lineskip` qui était avant la ligne modèle et on introduit `\LT@setprevdepth` pour fixer `\prevdepth` au début du prochain morceau, pour s'assurer que la glu `\lineskip` est ajoutée plus tard.

```
    \unskip
  \egroup}
```

`\LT@entry` Nous donnons ici la définition élémentaire de `\LT@entry`, c'est à dire celle utilisée dans les modèles d'alignement. Il ajoute un `\kern` seulement si le maximum est imposé par un morceau différent.

Le test `\ifhmode` donne la première entrée quand nous ne voulons pas ajouter un `&`.

```
\def\LT@entry#1#2{%
  \ifhmode\@firstofone{&}\fi\omit
  \ifnum#1=\c@LT@chunks
  \else
    \kern#2\relax
  \fi}
```

`\LT@entry@chop` Cette définition pour l'argument de `\LT@save@row` est utilisée pour mettre de côté tous ces maxima qui ne peuvent être vérifiés parce qu'ils apparaissent après la fin de la table. Cela peut se produire seulement si une table a été raccourcie (ou si l'ordre des lignes a été bousculé depuis la compilation précédente). Il est à noter que cela est prématuré : le dernier morceau est toujours à fixer et avec les limites tronquées.

```
\def\LT@entry@chop#1#2{%
  \noexpand\LT@entry
    {\ifnum#1>\c@LT@chunks
       1}{0pt%
     \else
       #1}{#2%
     \fi}}
```

`\LT@entry@write` Pour écrire une entrée dans le fichier **aux**, nous utilisons une définition, légèrement surprenante, qui a pour seul but d'éviter des lignes débordantes (qui pourraient dépasser les limites de TEX lors de la lecture du fichier **aux** ; vous aurez probablement besoin de quelques centaines de colonnes avant que cela ne se produise, mais...).

```
\def\LT@entry@write{%
  \noexpand\LT@entry^^J%
  \@spaces}
```

`\LT@kill` Cela termine le morceau courant comme ci-dessus, mais supprime deux lignes, la ligne modèle et la ligne tuée, avant de prendre en compte le morceau suivant. Depuis la V3.04, l'ancien morceau est « remis en boîte » au début de la boîte contenant le morceau suivant. Cela permet d'utiliser `\kill` dans les en-tête (de table) puisque cette commande doit être traitée dans une seule boîte.

```
\def\LT@kill{%
  \LT@echunk
  \LT@get@widths
  \expandafter\LT@rebox\LT@bchunk}
```

`\LT@rebox` Renvoie l'ancien morceau (box0) au début du nouveau morceau, ce qui supprime la ligne tuée. Cette macro a été ajoutée dans la V3.04.

```
\def\LT@rebox#1\bgroup{%
  #1\bgroup
  \unvbox\z@
  \unskip
  \setbox\z@\lastbox}
```

9.8 La Ligne modèle

La ligne modèle est prise en compte par `\LT@save@row`.

`\LT@blank@row` `\LT@build@blank`

Crée une ligne blanche si nous n'avons pas utililisé les informations du fichier `.aux`.

```
\def\LT@blank@row{%
  \xdef\LT@save@row{\expandafter\LT@build@blank
    \romannumeral\number\LT@cols 001 }}
```

Oups ! Qu'est-ce que c'est supposé être ? Un remplacement de la première tache de l'Appendice D du TEXbook. `\romannumeral` affecte à `\LT@cols` un `m` par unité suivi de `i`[7]. La macro ci-dessous remplace donc les `m` par les entrées adéquates.

```
\def\LT@build@blank#1{%
  \if#1m%
    \noexpand\LT@entry{1}{0pt}%
    \expandafter\LT@build@blank
  \fi}
```

`\LT@make@row`

Avant la version 4, par défaut on n'utilisait pas l'information du fichier `.aux` mais maintenant nous pouvons définir `\LT@make@row` pour que la commande utilise le fichier `.aux`, même pour les passes brouillon.

```
\def\LT@make@row{%
  \global\expandafter\let\expandafter\LT@save@row
    \csname LT@\romannumeral\c@LT@tables\endcsname
  \ifx\LT@save@row\relax
    \LT@blank@row
```

Là, c'est un cas légèrement difficile. Avant de décider de faire un modèle à partir du fichier `.aux`, nous vérifions que le nombre de champs est toujours le même. Si ce n'est pas le cas, soit le format de la table a changé, soit nous avons une table totalement mauvaise. Dans les deux cas, nous avons décidé de supprimer toutes informations accumulées et de redémarrer.

Le développement entre les `!`...`!` ci-dessous sera vide si le nombre de macros `\LT@entry` incluant les arguments dans `\LT@save@row` est égal à `\LT@cols`. S'il n'est pas vide, nous supprimons la ligne et nous repartons de zéro.

```
  \else
    {\let\LT@entry\or
     \if!%
         \ifcase\expandafter\expandafter\expandafter\LT@cols
         \expandafter\@gobble\LT@save@row
         \or
         \else
           \relax
         \fi
        !%
     \else
       \aftergroup\LT@blank@row
     \fi}%
```

[7] NdT : si le compteur vaut 12, `\number\LT@cols 001` vaut 12001, qui en chiffres romains donne `mmmmmmmmmmmmi`, soit un `m` par unité.

```
  \fi}
```

`\setlongtables` Redéfinit la macro `\LT@make@row` pour utiliser l'information du fichier `.aux`, s'il y a une ligne sauvegardée pour cette table avec le bon nombre de colonne.

Depuis la Version 3.02 longtable utilise le compteur interne `\c@LT@tables` au lieu du compteur table de LaTeX . Le message d'alerte a été ajouté dans la V3.04, ainsi que le `\global`, pour éviter de faire déborder la pile.

Depuis la version 4.01, `\setlongtables` ne fait rien puisqu'il n'est plus nécessaire, mais est défini comme `\relax` au bénéfice des anciens documents.

```
\let\setlongtables\relax
```

`\LT@get@widths` C'est le cœur de longtable. S'il n'y a pas de tête et pied de table, cette macro, associée avec la commande `\\` modifiée, pourrait tout à fait constituer une base pour un petit package simple pour les tables longues. Cette macro est trés inspirée de la macro `\endvrulealign` présentée en annexe D du TeXbook.

```
\def\LT@get@widths{%
```

`\global` ajouté par V3.04, pour éviter de faire déborder la pile.

Fait une boucle à travers la dernière ligne, supprimant la justification et sauvegardant les largeurs de boîte. Pour la V3.04, changement de la scratch box à 2 puisque la nouvelle commande `\kill` demande que `\box0` soit conservé.

```
  \setbox\tw@\hbox{%
    \unhbox\@ne
    \let\LT@old@row\LT@save@row
    \global\let\LT@save@row\@empty
    \count@\LT@cols
    \loop
      \unskip
      \setbox\tw@\lastbox
    \ifhbox\tw@
      \LT@def@row
      \advance\count@\m@ne
    \repeat}%
```

Rappelle la largeur si nous sommes dans le premier morceau.

```
  \ifx\LT@@save@row\@undefined
    \let\LT@@save@row\LT@save@row
  \fi}
```

`\LT@def@row` Ajoute une colonne à la ligne modèle. Nom changé (était `\defLT@save@row`) dans la Version 3, pour conserver la convention de dénommination `\LT@`.

```
\def\LT@def@row{%
```

Nous commençons par choisir les entrées respectives de nos anciennes lignes. Ces redéfinitions de `\LT@entry` sont limitées au groupe débuté en `\LT@get@widths`.

```
  \let\LT@entry\or
  \edef\@tempa{%
    \ifcase\expandafter\count@\LT@old@row
    \else
      {1}{0pt}%
    \fi}%
```

Maintenant, nous attachons la bonne définition devant `\LT@save@row` :

```
  \let\LT@entry\relax
  \xdef\LT@save@row{%
    \LT@entry
    \expandafter\LT@max@sel\@tempa
    \LT@save@row}}
```

`\LT@max@sel` Comment sélectionner la bonne combinaison. Noter que nous donnons l'information de l'ancien maximum, seulement si la taille ne change dans *aucune* direction. Si la taille a augmenté, nous aurons, bien sûr, un nouveau maximum. Si la taille a diminué, l'ancien maximum (qui était explicitement imposé à cause de l'existence du morceau courant) est non valide, et nous commençons avec ce morceau comme étant la nouvelle taille. Noter que même en cas d'égalité, nous *devons* utiliser la construction `\the\wd\tw@` au lieu de `#2`, parce que `#2` pourrait être lu à partir du fichier, et donc il pourrait exister des versions `\catcode` 11 de `p` et `t` dans lesquelles nous voulons les remplacer par leur versions `\catcode` 12 correctes.

```
\def\LT@max@sel#1#2{%
  {\ifdim#2=\wd\tw@
     #1%
   \else
     \number\c@LT@chunks
   \fi}%
  {\the\wd\tw@}}
```

9.9 La commande `\hline`

`\LT@hline` `\hline` et `\hline\hline` produisent tous deux *deux* lignes, la seule différence étant la justification et les « penalties ». C'est pourquoi un saut de page au niveau d'une `\hline` produit une ligne sur les deux pages à la fois.[8]. De plus, cette `\hline` est plus comparable à `\cline{1-\LT@cols}`. Les `\hline` de tabular devraient tracer des lignes de la largeur totale de la page.

```
\def\LT@hline{%
  \noalign{\ifnum0=`}\fi
    \penalty\@M
    \futurelet\@let@token\LT@@hline}
```

`\LT@@hline` Ce code est basé sur celui de `\cline`. Deux copies de la ligne sont produites, comme décrit ci-dessus.

```
\def\LT@@hline{%
  \ifx\@let@token\hline
    \global\let\@gtempa\@gobble
    \gdef\LT@sep{\penalty-\@medpenalty\vskip\doublerulesep}%
  \else
    \global\let\@gtempa\@empty
    \gdef\LT@sep{\penalty-\@lowpenalty\vskip-\arrayrulewidth}%
```

[8] longtable a toujours réalisé cela, mais peut-être ce serait mieux si les hlines étaient *omises* au niveau du saut de page, puisque habituellement, les tête et pied de table mettent une hline ici, de toutes façns

```
  \fi
  \ifnum0=`{\fi}%
  \multispan\LT@cols
     \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
  \noalign{\LT@sep}%
  \multispan\LT@cols
     \unskip\leaders\hrule\@height\arrayrulewidth\hfill\cr
  \noalign{\penalty\@M}%
  \@gtempa}
```

9.10 Légendes

`\LT@caption` La légende est donnée par `\multicolumn{\LT@cols}{c}{`⟨*une parbox avec la légende de la table*⟩`}`

```
\def\LT@caption{%
  \noalign\bgroup
    \@ifnextchar[{\egroup\LT@c@ption\@firstofone}\LT@capti@n}
```

`\LT@c@ption` Commande de légende (avec [argument optionnel]). `\protect` ajouté dans la version 3. `\fnum@table` ajouté dans la V3.05.

```
\def\LT@c@ption#1[#2]#3{%
  \LT@makecaption#1\fnum@table{#3}%
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else
     {\let\\\space
     \addcontentsline{lot}{table}{\protect\numberline{\thetable}{#2}}}%
  \fi}
```

`\LT@capti@n` Commande de légende (sans [argument optionnel])

```
\def\LT@capti@n{%
  \@ifstar
    {\egroup\LT@c@ption\@gobble[]}%
    {\egroup\@xdblarg{\LT@c@ption\@firstofone}}}
```

`\LT@makecaption` Met la légende dans une boîte de largeur 0pt, pour que ça n'affecte jamais les largeurs de colonne. La légende comprend une `\parbox` de largeut `\LTcapwidth`.

```
\def\LT@makecaption#1#2#3{%
  \LT@mcol\LT@cols c{\hbox to\z@{\hss\parbox[t]\LTcapwidth{%
```

Basée sur `\@makecaption` de la classe article, `#1` est `\@gobble` dans la forme `*`, et `\@firstofone` autrement.

```
    \sbox\@tempboxa{#1{#2: }#3}%
    \ifdim\wd\@tempboxa>\hsize
      #1{#2: }#3%
    \else
      \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
    \fi
    \endgraf\vskip\baselineskip}%
  \hss}}}
```

9.11 La routine de sortie

La méthode utilisée ici pour relier une routine de sortie, utilisée dans un but spécial, à la routine standard de LaTeX provient en droite ligne du package multicol de F. Mittelbach

\LT@output De fait, ce n'est pas si terrible, FM ayant déjà dégagé le terrain.

```
\def\LT@output{%
  \ifnum\outputpenalty <-\@Mi
    \ifnum\outputpenalty > -\LT@end@pen
```

Si c'était un flottant ou une marge de paragraphe que nous réclamons.

```
      \LT@err{floats and marginpars not allowed in a longtable}\@ehc
    \else
```

Nous atteignons la fin de la table, au moins par défilement,

```
      \setbox\z@\vbox{\unvbox\@cclv}%
      \ifdim \ht\LT@lastfoot>\ht\LT@foot
```

Le dernier pied peut ne pas correspondre, donc : [9]

```
        \dimen@\pagegoal
        \advance\dimen@-\ht\LT@lastfoot
        \ifdim\dimen@<\ht\z@
          \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
          \@makecol
          \@outputpage
          \setbox\z@\vbox{\box\LT@head}%
```

Fin de `\ifdim\dimen@<\ht\z@`[10].

```
        \fi
```

Fin de `\ifdim \ht\LT@lastfoot > \ht\LT@foot`.

```
      \fi
```

Réinitialisation de `\@colroom`.

```
      \global\@colroom\@colht
      \global\vsize\@colht
```

Met la dernière page de la table dans la liste verticale principale.

```
      \vbox
        {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
```

Fin de `\ifnum\outputpenalty > -\LT@end@pen`.

```
    \fi
```

Fin de `\outputpenalty > -\@Mi`.

```
  \else
```

[9] Une alternative pourraît être la suppression d'un petit bout du dernier morceau, de façon à ce que la dernière page n'ait pas de sections de tête et de pied, mais il est difficile de réaliser cela d'une manière robuste.

[10] NdT : La documentation originale dit ici « End of `\ifdim\dimen@<\ht\@cclc` » qui semble être une faute, puisque le `\if` effectivement fermé porte sur la hauteur de la boîte 0 et non pas 255.

Si nous n'avons pas atteint la fin de la table,

```
    \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
    \@makecol
    \@outputpage
```

Réinitialisation de `\vsize`.

```
      \global\vsize\@colroom
```

Met la tête de table en haut de la page suivante.

```
    \copy\LT@head
```

Fin de `\ifnum\outputpenalty <-\@Mi`.

```
  \fi}
```

9.12 Commande pour les en-tête et pied de table

`\LT@end@hd@ft` Le cœur de `\endhead` et compagnie. Stocke le morceau courant dans la boîte spécifiée par `#1`. Produit une erreur si la table est déjà commencée. Puis, commence un nouveau morceau.

```
\def\LT@end@hd@ft#1{%
  \LT@echunk
```

Changement de `\relax` en `\endgraf` pour V3.04, *cf* `\LT@start`.

```
  \ifx\LT@start\endgraf
    \LT@err
     {Longtable head or foot not at start of table}%
     {Increase LTchunksize}%
  \fi
  \setbox#1\box\z@
  \LT@get@widths
  \LT@bchunk}
```

`\endfirsthead` `\endhead` `\endfoot` `\endlastfoot` Appelle `\LT@end@hd@ft` avec la boîte adéquate.

```
\def\endfirsthead{\LT@end@hd@ft\LT@firsthead}
\def\endhead{\LT@end@hd@ft\LT@head}
\def\endfoot{\LT@end@hd@ft\LT@foot}
\def\endlastfoot{\LT@end@hd@ft\LT@lastfoot}
```

9.13 La commande `\multicolumn`

Les plus anciennes versions nécessitaient une forme brouillon spéciale de `\multicolumn`. Cela n'est plus nécessaire dans la version 4, et donc ces commandes ont été supprimées.

`\LTmulticolumn`

`\LT@mcwarn`

9.14 Notes de bas de page

La commande `\footnote` standard fonctionne dans une colonne `c`, mais nous avons besoin de modifier cette définition pour une colonne `p`, pour passer outre le niveau supplémentaire d'emboîtement. Ces macros sont basées sur le package array, mais devraient être OK pour l'environnement tabular standard.

`\LT@startpbox` Ajoute un code supplémentaire pour changer la définition de `\@footnotetext`.

```
\def\LT@startpbox#1{%
  \bgroup
    \let\@footnotetext\LT@p@ftntext
    \setlength\hsize{#1}%
    \@arrayparboxrestore
    \vrule \@height \ht\@arstrutbox \@width \z@}
```

`\LT@endpbox` Après la fermeture de la parbox, développement de `\LT@p@ftn` qui exécutera une série de
`\footnotetext[`⟨*num*⟩`]{`⟨*note*⟩`}`
Après avoir été enlevées de la parbox, elles peuvent aller à leur place à partir d'ici.

```
\def\LT@endpbox{%
  \@finalstrut\@arstrutbox
  \egroup
  \the\LT@p@ftn
  \global\LT@p@ftn{}%
  \hfil}
```

`\LT@p@ftntext` Dans la colonne `p`, sauvegarde juste le texte de la note de bas de page dans registre à jeton.

```
\def\LT@p@ftntext#1{%
  \edef\@tempa{\the\LT@p@ftn\noexpand\footnotetext[\the\c@footnote]}%
  \global\LT@p@ftn\expandafter{\@tempa{#1}}}%
⟨/package⟩
```

Historique des changements