

Tutoriel pour l'utilisation du package *PackageTheatre.sty*

Ludovic Ducobu

25 & 26 juillet 2016

Dernière modification : 29 juin 2017

Table des matières

1.	Introduction	1
2.	Installation et utilisation	1
2.1.	Installation	1
2.2.	Utilisation	1
3.	Les bases de <i>PackageTheatre.sty</i>	2
3.1.	L'environnement <i>piece</i>	2
3.2.	La commande <i>titrepiece</i>	2
3.3.	L'environnement <i>acte</i>	3
3.4.	La commande <i>scene</i>	4
3.5.	La commande <i>replique</i>	5
3.6.	La commande <i>didascalie</i>	6
3.7.	La commande <i>CreePersonnage</i>	6
4.	Les options	8
4.1.	Options d'espacement des scènes	8
4.2.	Options d'entête	9
4.3.	Options pour la table des matières	11
5.	conclusion	11

1. Introduction

L^AT_EX est un outil formidable pour les maniaques de typographie. Il offre un grand nombre d'outils permettant de définir simplement¹ et de manière automatisée la mise en page d'un document.

Toute la puissance de L^AT_EX se révèle lorsque l'on prend le peine de se familiariser avec les *packages*. Les packages sont des fichiers annexes dans lesquels on peut définir une fois pour toute un certain nombre de commandes qui pourront être utilisées dans n'importe quel fichier “.tex” pour autant que l'on y appelle le package au moyen de la commande “\usepackage{NomDuPackage}” dans le préambule du document “.tex”.

Dans ce tutoriel nous présentons les différents objets définis dans le package “*PackageTheatre.sty*”. Ce package a été développé par Ludovic DUCOBU entre mars et juillet de l'année 2016 afin de donner les outils nécessaires à la rédaction d'une pièce de théâtre avec L^AT_EX.

Dernière modification du package : 29 juin 2017

2. Installation et utilisation

2.1. Installation

Pour pouvoir utiliser le package, il suffit de disposer du fichier “*PackageTheatre.sty*” et de le placer dans le même dossier que le fichier “.tex” dans lequel il est appelé². Il suffit alors d'entrer

```
\usepackage{PackageTheatre}
```

dans le préambule du document pour disposer de tous les outils du package.

2.2. Utilisation

Le package a été écrit pour que les commandes soient utilisées au sein d'un document de la classe “*report*” ou “*book*” il est donc vivement conseillé de n'utiliser le package que dans des documents commençant par

```
\documentclass{report}
ou
\documentclass{book}.
```

Le package a été écrit pour la rédaction de pièces de théâtre en français et, par conséquent, “*PackageTheatre.sty*” importe automatiquement les packages *inputenc* avec l'option “utf8” et *babel* avec l'option “français”.

Les packages *amsmath*, *amsfonts*, *amssymb*, *amsthm*, *braket*, *graphicx* et *color* sont également importés pour les petits malins qui voudraient écrire des textes étranges remplis de symboles mathématiques et/ou avec des indications en couleur.

Pour des raisons pratiques (voir les options d'entête ci après) le package *fancyhdr* est également importé.

Dès lors, sauf demande spéciale, vous ne devriez avoir besoin d'importer que “*PackageTheatre.sty*” pour la rédaction de votre pièce.

Le seul package dont vous pourriez avoir besoin et qui ne soit pas importé d'office lors de l'importation de “*PackageTheatre.sty*” est le package *geometry* qui vous permet de redéfinir les marges de votre document. Ce package a volontairement été ignoré afin de laisser l'utilisateur libre de définir lui même les marges de son document.

1. Pour qui a des bases en programmation

2. Si L^AT_EX est correctement installé, on peut aussi le placer dans l'arborescence de L^AT_EX pour pouvoir l'appeler dans n'importe quel dossier. Il faut reconnaître que placer le fichier dans le même dossier que le fichier “.tex” est une méthode d'installation en soi plus simple mais un peu “moins propre” sur le plan de la programmation. Cependant cette méthode fonctionnera toujours.

3. Les bases de *PackageTheatre.sty*

Dans cette section, nous présentons les outils du package.

3.1. L’environnement *piece*

Cet environnement est le plus important du package. C’est lui qui définit ce qui dans le fichier “*.tex*” est dans le texte de la pièce à proprement parler et ce qui ne l’est pas. Son utilisation est la même que pour tout environnement :

```
\begin{piece}  
    Le texte de la pièce...  
\end{piece}
```

L’ouverture d’un environnement *piece* s’accompagnera automatiquement d’un saut de page.

À ce stade une remarque s’impose : Nous avons volontairement choisis de définir la pièce comme un environnement et non de définir une nouvelle classe de document. Le but de cette manœuvre était de faire en sorte que le package offre l’opportunité d’insérer un nombre arbitraire de pièces différentes au sein d’un seul fichier “*.tex*”³.

3.2. La commande *titrepiece*

ATTENTION : Cette commande est en fait une “sous-commande” de l’environnement *piece*, elle ne peut donc être appelée qu’à l’intérieur dudit environnement !

Cette commande permet de définir le titre de la pièce en cours. Elle est à utiliser à l’endroit où l’on veut que le titre s’affiche (il cependant conseillé de mettre le titre dès l’ouverture de l’environnement *piece*).

La commande prend évidemment un argument : Le titre de la pièce en cours. En entrant le code :

```
\titrepiece{La nuit tous les chats sont gris}  
Le texte...
```

on obtiendra dans le fichier compilé :

La nuit tous les chats sont gris

Le texte...

Notons que le fait de devoir utiliser une autre commande que `\title` pour définir le titre d’une pièce provient de notre souhait de permettre à l’utilisateur d’inclure plusieurs pièces au sein d’un même fichier “*.tex*”. La commande `\title` définit le titre du document “*.tex*” complet tandis que *titrepiece* définit localement le titre de la pièce se trouvant à l’intérieur de l’environnement *piece* en cours. Si vous incluez différentes pièces au sien d’un même fichier “*.tex*”, vous pourrez ainsi répéter l’utilisation de la commande *titrepiece* dans chaque environnement *piece* sans qu’il n’y ai de contradiction entre les pièces ni avec le titre du document “*.tex*” (qui pourrait être “Recueil de mes pièces”⁴).

3. Pour les utilisateurs souhaitant regrouper plusieurs de leur pièces au sein d’un même ouvrage tout en conservant une numérotation des pages cohérente, par exemple.

4. ou quelque chose d’approchant...

3.3. L'environnement *acte*

ATTENTION : Cet environnement est en fait un “sous-environnement” de l’environnement *piece*, il ne peut donc être appelé qu’à l’intérieur dudit environnement !

Comme son nom le laisse paraître, l’environnement *acte* permet de délimiter les différents actes d’une pièce donnée. Il s’appelle comme tout environnement :

```
\begin{acte}  
  Le texte de l’acte...  
\end{acte}
```

L’ouverture d’un acte donne lieu à un saut de page et à l’apparition de

Acte 1

Le texte de l’acte...

après compilation du fichier “*.tex*”.

Lorsque l’on utilise plusieurs environnements *acte* dans une même pièce, ceux-ci se numérotent automatiquement. Le code suivant :

```
\begin{acte}  
  Le texte de l’acte 1...  
\end{acte}  
  
\begin{acte}  
  Le texte de l’acte 2...  
\end{acte}
```

donnera lieu à l’affichage suivant dans le texte final (avec à chaque fois l’ouverture d’une nouvelle page au début de chaque acte) :

Acte 1

Le texte de l’acte 1...

Acte 2

Le texte de l’acte 2...

et ainsi de suite...

3.4. La commande *scene*

ATTENTION : Cette commande est en fait une “sous-commande” de l’environnement *piece*, elle ne peut donc être appelée qu’à l’intérieur dudit environnement !

Cette commande permet de définir le début des différentes scènes d’une pièce donnée. Lors de l’appel de cette commande via

```
\scene  
le texte de la scène...
```

vous obtiendrez après compilation :

Scène I

le texte de la scène...

Lorsque cette commande est utilisée plusieurs fois de suite, les scènes se numérotent automatiquement avec un numéro en chiffre romain. Le code :

```
\scene  
le texte de la scène 1...  
\scene  
le texte de la scène 2...  
\scene  
le texte de la scène 3...
```

donnera après compilation :

Scène I

le texte de la scène 1...

Scène II

le texte de la scène 2...

Scène III

le texte de la scène 3...

Si votre pièce comporte des actes, les scènes se numérotent automatiquement en fonction des actes. Le code :

```
\begin{acte}  
  \scene  
  Une scène...  
  \scene  
  \scene  
\end{acte}
```

```
\begin{acte}  
  \scene  
  \scene  
\end{acte}
```

donnera⁵ après compilation du fichier “.tex” :

Acte 1

Scène I

Une scène...

Scène II

Scène III

Acte 2

Scène I

Scène II

3.5. La commande *replique*

ATTENTION : Cette commande est en fait une “sous-commande” de l’environnement *piece*, elle ne peut donc être appelée qu’à l’intérieur dudit environnement !

REMARQUE IMPORTANTE : L’utilisation de la commande *replique* a été rendue obsolète par l’ajout de la commande *CreePersonnage* (voir ci-dessous) cependant nous continuons de décrire le comportement de cette commande afin d’être le plus complet possible et de pouvoir expliquer plus clairement le fonctionnement de la commande *CreePersonnage* en elle-même.

Pouvoir numéroter les différentes scènes et actes de son texte c’est bien mais pouvoir remplir lesdites scènes et actes, c’est mieux. La commande *replique* permet d’avoir un style uniforme pour l’encodage des répliques des personnages tout au long de la pièce.

Cette commande prends deux arguments : Le nom du personnage qui parle et sa réplique. Par exemple le code :

```
\replique{Marc}{Bonjour.}
\replique{Mathieu}{Salut !}
\replique{Marc}{Beau temps aujourd’hui...}
```

donnera :

MARC : Bonjour.

MATHIEU : Salut !

MARC : Beau temps aujourd’hui...

ASTUCE : En définissant dans le préambule des commandes \Marc et \Mathieu comme

5. Les actes seront sur des pages différentes mais nous montrons ici le résultat sur une même page dans un soucis de gain de place.

```
\newcommand{\Marc}[1]{\replique{Marc}{#1}}
\newcommand{\Mathieu}[1]{\replique{Mathieu}{#1}}
```

vous gagnerez du temps et de la clarté lors de la rédaction puisque vous pourrez écrire

```
\Marc{Bonjour.}
\Mathieu{Salut !}
\Marc{Beau temps aujourd'hui...}
```

pour obtenir le résultat ci-dessus après compilation.

3.6. La commande *didascalie*

ATTENTION : Cette commande est en fait une “sous-commande” de l’environnement *piece*, elle ne peut donc être appelée qu’à l’intérieur dudit environnement !

Dans toute pièce qui se respecte, en plus du texte (les répliques) on trouve également un certain nombre d’informations en sous-texte sous la forme des fameuses didascalies.

La commande bien nommée *didascalie* prends un argument : le texte de la didascalie. Le code :

```
\didascalie{Une remarque en passant...}
```

donnera lieu à :

(*Une remarque en passant...*)

Un exemple plus complet :

```
\didascalie{À l’ouverture du rideau, allumer la lumière.}

\Marc{\didascalie{Sur un ton maussade :) Bonjour.}
\Mathieu{Salut !}
\Marc{Beau temps aujourd’hui...}
```

donnera :

(*À l’ouverture du rideau, allumer la lumière.*)

MARC : (*Sur un ton maussade :) Bonjour.*

MATHIEU : Salut !

MARC : Beau temps aujourd’hui...

3.7. La commande *CreePersonnage*

La commande

Cette commande a été développée pour sublimer l’utilisation de la commande *replique*. Comme nous l’avons remarqué un peu plus haut, la commande *replique*, bien que très intuitive, présente un “inconvenient” majeur : dans un texte donné, on est amené à faire parler à plusieurs reprises un certain nombre de personnages, toujours les mêmes (les protagonistes de la pièce). Le fait que la commande *replique* requière de manière systématique le nom du personnage qui s’apprête à parler est “désagréable” en ce sens qu’il est plus élégant de définir une commande qui ne prend qu’un argument (le texte de la réplique) et qui affiche automatiquement le nom du personnage choisis (cela évite de commettre une erreur d’inattention en recopiant le nom du personnage pour la nième fois). C’est ce que nous proposons en astuce dans le paragraphe décrivant la commande *replique*. Pour vous éviter d’avoir à faire cette étape par vous même et rendre votre code plus lisible (mais pas seulement !) nous avons développé la commande *CreePersonnage*.

Cette commande prend deux arguments et construit pour vous tous les outils nécessaires pour faire s'exprimer ou pour faire référence à un personnage donné. Le premier argument de la commande est le nom du personnage que l'on est en train de définir. Le second est le nom de la commande par laquelle on fera appel à ce personnage (ces deux arguments sont a priori différents, on peut avoir un personnage au nom très long que l'on souhaite appeler au moyen d'une commande plus courte).

Par exemple, en écrivant

$$\backslash\text{CreePersonnage}\{\text{Mathieu}\}\{\text{mat}\}$$

on crée un personnage nommé Mathieu mais que l'on appellera au moyen de la commande `\mat`. Après avoir utilisé *CreePersonnage*, une commande a donc été créée (`\mat` dans notre exemple). Cette commande est particulière car elle peut prendre ZÉRO OU UN argument.

Si on ne lui donne aucun argument, la commande considère que l'on est en train de faire référence au personnage (dans une didascalie par exemple). Dans ce cas la commande se contentera d'afficher le nom du personnage.

Si on lui donne un argument, la commande considère que l'on est en train d'ouvrir une réplique du personnage concerné. Dans ce cas le comportement de la commande sera essentiellement celui de la commande *replique* à laquelle on aurait donné le nom du personnage et le texte de la réplique (l'argument donné à la commande générée par *CreePersonnage*).

Par exemple, pour un personnage créé comme décrit ci-dessus, le code suivant

$$\begin{aligned} &\backslash\text{mat}\{\text{C'est bien compliqué toutes ces commandes}\backslash\text{dots}\} \\ &\backslash\text{didascalie}\{\backslash\text{mat se gratte la tête.}\} \end{aligned}$$

donnera après compilation

MATHIEU : C'est bien compliqué toutes ces commandes ...

(MATHIEU *se gratte la tête.*)

L'AVANTAGE : L'avantage de cette manière de faire est que l'on définit une fois pour toute le nom du personnage. Plus de risque de se tromper en recopiant ET, si d'aventure vous décidiez de changer le nom d'un de vos personnages en cours d'écriture, vous n'auriez qu'à le changer une seule fois au niveau de la création du personnage pour qu'il soit modifié automatiquement dans tout le texte.

Et, fous de L^AT_EX que nous sommes, on ne s'arrête pas là !

En bonus !

Une "manie" assez répandue chez les comédiens consiste à compter le nombre de répliques du personnage dont ils ont hérité. Et, pour satisfaire cette curiosité dévorante, nos ancêtres en étaient réduits à compter manuellement. Mais grâce à la commande *CreePersonnage* ce temps-là est révolu !

En plus de la commande permettant de faire référence à un personnage donné, *CreePersonnage* crée un compteur, portant le même nom que la commande permettant de référer au personnage, qui comptera le nombre de répliques du personnage. Comme toujours avec les compteurs L^AT_EX, on affichera le nombre stocké dans le compteur en entrant `\theNomDeLaCommande`.

Dans notre exemple, on écrira `\themat` pour savoir combien de répliques ont été prononcées par Mathieu jusque là. En utilisant la commande à la fin de la pièce, on aura ainsi naturellement accès au nombre total de répliques que Mathieu.

Et, tant qu'à y être, nous avons également prévu un compteur "totalrepliques" comptant le nombre total de répliques de la pièce en cours. En tapant `\thetotalrepliques`, on aura donc accès au nombre de répliques tous personnages confondus jusqu'à l'endroit où la commande est appelée. En l'utilisant à la fin de la pièce, ici encore, on connaîtra le nombre total de répliques de sa prose.

REMARQUE : Pour plus de lisibilité de votre code, l'idéal est de mettre tous les appels à la commande *CreePersonnage* au même endroit, par exemple dans le préambule⁶ de votre document *“.tex”*.

CEPENDANT, si le document *“.tex”* que vous écrivez regroupe plusieurs pièces il vaut mieux mettre les appels à la commande *CreePersonnage* pour une pièce donnée juste avant de commencer votre pièce⁷. Cette manoeuvre permet de remettre à zéro les compteurs de répliques pour les personnages apparaissant dans plusieurs pièces, et ce afin d'éviter d'obtenir un compte des répliques erroné. Dans notre exemple, il faudrait donc remettre `\CreePersonnage{Mathieu}{mat}` au début de chaque pièce où le personnage de Mathieu apparaît même si le personnage avait déjà été créé pour une pièce précédente. D'une certaine façon on peut considérer que ce nouvel appel à la commande *CreePersonnage* “réinitialise” le personnage.

Si vous souhaitez définir des compteurs indépendants pour les différentes apparitions d'un personnage au cours de différentes pièces (sans devoir le réinitialiser) il convient de définir des commandes d'appel au personnage différentes pour chaque pièce. Par exemple, utiliser

`\CreePersonnage{Mathieu}{mat}`

pour une première pièce et

`\CreePersonnage{Mathieu}{mat2}`

pour une seconde. De cette façon, les compteurs “mat” et “mat2” seront indépendants mais l'appel à l'une ou à l'autre des commandes permettra de faire référence à Mathieu.

N.B : Le compteur “totalrepliques” est automatiquement remis à zéro lors de l'ouverture d'une nouvelle pièce. Il sert donc de compteur du nombre total de répliques pour la pièce en cours et non de compteur du nombre total de répliques toutes pièces confondues si jamais votre document *“.tex”* en comporte plusieurs différentes.

4. Les options

À ce stade du tutoriel, vous disposez de tous les outils nécessaires à la rédaction de votre pièce. Cependant, dans un souci de flexibilité du package, un certain nombre d'options de personnalisation du style de votre texte ont été implémentées. Ce sont ces options que nous présentons dans cette section.

4.1. Options d'espacement des scènes

Comme nous l'avons mentionné ci-avant, l'ouverture d'un acte s'accompagne automatiquement d'un “saut de page” c.à.d. que le début d'un acte se situe toujours au début d'une nouvelle page. Par défaut cela n'est pas le cas pour les scènes. Lors de l'appel de la commande *scene*, des espacements sont ajoutés mais l'on continue sur la page en cours (pour autant qu'il reste suffisamment de place).

La commande *ActiveEspaceScene*

Pour ceux que cela dérangerait, pour ceux qui souhaitent espacer leur texte un maximum, pour ceux là⁸ nous avons développé une commande très simple : *ActiveEspaceScene*. L'appel de cette commande changera le comportement du package de telle sorte que toutes les scènes (sauf la première afin d'éviter de se retrouver avec une page comportant uniquement le titre de la pièce et/ou de l'acte) débiteront sur une nouvelle page.

6. c.à.d. avant le `\begin{document}`

7. c.à.d. juste avant le `\begin{piece}`

8. et pour tous les autres

La commande *ArreteEspaceScene*

Vous êtes heureux de pouvoir aérer votre texte seulement le sort s’acharne : Une de vos scènes se termine après seulement deux ou trois lignes sur une nouvelle page de telle sorte que que votre texte comporte une page presque vide (ce qui n’est pas génial du point de vue de l’économie de papier). Que faire ? Pas de panique, la commande *ArreteEspaceScene* a été faite pour vous !

Comme son nom le laisse présager, cette commande annule les effets de la commande *ActiveEspaceScene* pour tout le texte situé en dessous d’elle (à moins que l’on ne vienne réinsérer une commande *ActiveEspaceScene*) de telle sorte que toutes les scènes suivantes (jusqu’à un nouvel appel de la commande *ActiveEspaceScene*) apparaîtront sans qu’un saut de page soit effectué.

Notons que l’appel de la commande *ArreteEspaceScene* sans qu’une commande *ActiveEspaceScene* n’ait été appelée au préalable ne provoquera aucune erreur de compilation. Cela sera simplement sans effet.

La combinaison de ces deux commandes vous permettra ainsi de faire en sorte que votre code soit aéré juste ce qu’il faut sans devoir inscrire explicitement `\pagebreak` au début de 99 scènes sur 100.

Remarquons, pour terminer, que nous avons volontairement omis de définir des commandes analogues pour l’espacement des actes car, à notre sens, un acte est à une pièce ce qu’un chapitre est à un roman. Il s’accompagne donc toujours d’un saut de page.

4.2. Options d’entête

L’allure de votre pièce vous plait déjà beaucoup mais vous aimeriez avoir un repère en haut de page pour vous dire à quelle scène de quel acte appartient la page que vous êtes en train de lire et vous êtes trop fainéant pour remonter quelques pages plus haut pour retrouver le début de la scène en cours ? Pas de panique, l’équipe de “*PackageTheatre.sty*” a pensé à tout⁹ !

Les commandes pour définir et personnaliser l’entête sont basées sur les outils développés dans le package *fancyhdr*. C’est pourquoi ce package est importé lors de l’appel de “*PackageTheatre.sty*”.

La commande *RefCoinPage*

Par défaut, rien n’apparaît en entête de vos pages ; la commande *RefCoinPage* est là pour changer ça. L’appel de cette commande changera le comportement du package : Une fois cette commande appelée¹⁰, vous verrez apparaître en haut de la page l’acte et la scène en cours (la scène en cours seulement si la pièce ne comporte pas d’acte).

Si votre document est défini de telle sorte que les pages de gauche (pages impaires) et les pages de droite (pages paires) sont traitées différemment¹¹, la référence de la scène en cours se trouvera sur la droite des pages paires et sur la gauche des pages impaires par défaut.

Dans le cas contraire (toutes les pages étant alors traitées comme des pages impaires), la référence apparaîtra par défaut à gauche.

La commande *auteurpiece*

Maintenant que vous avez une référence à la scène en cours dans un coin supérieur de votre page, vous aimeriez peut-être mettre une information dans l’autre coin (votre nom par exemple pour revendiquer fièrement la paternité de votre œuvre), c’est là que la commande *auteurpiece* intervient.

Cette commande prend un argument. Lorsque vous l’aurez appelée, l’argument que vous y aurez placé apparaîtra systématiquement dans le coin supérieur opposé à celui où l’a référence à la scène en cours apparaît.

Il est important de noter que, contrairement à la commande *titrepiece*, cette commande est définie **globalement**. Autrement dit, elle peut être appelée à l’extérieur de l’environnement pièce. Cela fait que la

9. Du moins ça on y a pensé...

10. Pour garantir l’uniformité du texte, nous conseillons d’inclure cette commande dans le préambule du fichier “*.tex*”.

11. C’est le cas par défaut avec la classe *book* ou avec toute autre classe pour autant que l’on spécifie l’option *twoside* lorsque l’on définit la classe du document dans la première ligne du code.

référence entrée en argument apparaîtra sur toutes les pages du document final (même celles situées hors de la pièce).

Soucis : Vous voulez inscrire quelque chose dans le coin supérieur de votre page mais ce n'est pas votre nom et vous ne voulez pas la même chose partout.

Solution : Il vous suffit d'appeler la commande *auteurpiece* une nouvelle fois avec un autre argument pour changer la référence sur toutes les pages situées en dessous de votre commande (avant éventuellement un nouvel appel de la commande). De cette façon, vous pouvez avoir ce que vous voulez en entête de votre page sur les pages que vous voudrez dans ou hors de l'environnement *piece*. Le nom de la commande est simplement le reflet du fait que nous l'avons initialement conçue pour indiquer le nom de l'auteur partout mais rien de vous empêche de personnaliser encore plus vos entête.

Soucis : Vous ne voulez rien dans le coin dédié au nom de l'auteur sur certaines pages.

Solution : Appelez la commande avec rien entre les crochets délimitant l'argument.

La commande *InverseSensRef*

Vous avez défini une jolie entête pour votre texte mais vous auriez préféré que la référence à la scène soit à droite et le nom de l'auteur à gauche au lieu de l'inverse ? Vous êtes décidément bien capricieux... Mais qu'importe, nous avons anticipé ce caprice avec la commande *InverseSensRef* !

En appelant cette commande¹², le coin assigné à la scène en cours et le coin assigné au nom de l'auteur seront systématiquement intervertis partout dans le document.

NOTE : Un nouvel appel de la commande **ne** rétablit **pas** l'ancienne configuration (en revanche effacer la commande si, bien entendu).

Les commandes *EffaceRefScene* et *EffaceRefTitre*

Définir une entête vous plaît mais avoir le titre de la scène en cours ne vous dit rien ? La commande *EffaceRefScene* est littéralement faite pour vous ! En appelant cette commande, l'entête arrêtera d'afficher la scène en cours. Par défaut, le package placera le titre de la pièce (celui défini via la commande *titrepiece*) à la place.

Vous ne voulez pas du titre de la pièce non plus ? ... Bon d'accord mais après c'est fini les caprices!!! Si vraiment vous ne voulez rien en entête à part ce que vous avez défini via la commande *auteurpiece*, la commande *EffaceRefTitre* effacera le titre de la pièce de l'entête.

Comprenons nous bien : En utilisant successivement ces deux commandes, il n'y aura plus rien dans le coin supérieur initialement dévolu à la scène en cours cependant l'option d'entête en elle même sera toujours active (la commande *RefCoinPage* ayant *a priori* été appelée) et la référence définie grâce à *auteurpiece* sera toujours visible. Si vraiment vous ne voulez rien de rien du tout en entête... Ben fallait juste pas utiliser *RefCoinPage*...

Remarque

Ces cinq commandes ont été définies avec l'aide des outils du package *fancyhdr* afin de simplifier au maximum la vie de l'utilisateur de "*PackageTheatre.sty*". L'utilisateur pour qui cette flexibilité n'est pas encore suffisante est invité à se familiariser directement avec le package *fancyhdr*.

Ce package permet de personnaliser l'entête de son document autant qu'on peut en rêver. Si nous avons défini ces quelques options d'entête c'est pour permettre aux utilisateurs qui ne sont pas familiers avec ce package d'avoir des raccourcis pour personnaliser un minimum l'entête de leur document. Cependant nous avons jugé bon de nous arrêter avant d'avoir purement et simplement dupliqué les outils offerts par *fancyhdr* en essayant de rendre notre package le plus flexible possible.

C'est pourquoi nous invitons chaleureusement l'utilisateur souhaitant définir une entête tout à fait particulière à apprendre à se servir de *fancyhdr*.

12. de préférence dans le préambule

4.3. Options pour la table des matières

Maintenant que vous savez comment rédiger votre pièce et la personnaliser (presque) suivant la moindre de vos fantaisies, la seule chose sur laquelle vous pourriez encore vouloir de l'influence c'est la table des matières du document.

Par défaut, le titre de la pièce est affiché dans la table des matières ainsi que les éventuels actes et scènes qui structurent le texte.

La commande *EffaceSceneTableMatiere*

permet d'enlever la position des différentes scènes de la table des matières tout en laissant la position des éventuels actes.

La commande *EffaceActeTableMatiere*

permet d'enlever les actes éventuels et les scènes de la table des matières¹³ pour ne laisser que le titre de la pièce.

Cette fois encore, c'est volontairement que nous n'avons défini aucune option permettant de retirer le titre de la pièce de la table des matières. Si vous voulez vraiment que rien n'apparaisse dans la table des matières, autant ne pas en mettre dans le document...

5. conclusion

Ici s'achève le tutoriel pour l'utilisation du fichier "*PackageTheatre.sty*". Nous espérons sincèrement que les outils développés dans ce package vous seront autant utiles qu'agréables à utiliser.

Pour toute suggestion d'amélioration, remarque, bugg, mouvement d'humeur ou autre n'hésitez pas à nous contacter via l'adresse mail ludoducobu@hotmail.com en spécifiant en objet "*PackageTheatre.sty* : l'objet de votre mail".

Il ne nous reste plus pour terminer ce tutoriel qu'à vous souhaiter une bonne route et de nombreuses créations artistiques avec "*PackageTheatre.sty*" !!!

13. il serait aberrant d'enlever les actes mais pas les scènes.