

Manuel de référence pour Almost Free Text (AFT)

Todd A. Coram

May 22, 2010

Version 5.97

révisée le 08/23/2009

traduction le 22/05/2010

Trad. Fr. Christophe Masutti ¹ (Linux et les Choses <http://linuxetleschoses.tuxfamily.org>)



Contents

1 Introduction	1
1.1 Comment lire ce document	2
2 Lancer AFT	2
3 AFT Text: Un exemple	3
4 Revue de la syntaxe	4
4.1 Continuité des lignes	4
4.2 Les modes	5
4.3 Caractères spéciaux	5
4.4 Filtrage des lignes	5

¹christophePOINTmasuttiINSERT_AT_HEREgmailPOINTcom

<i>CONTENTS</i>	2
5 Sections	5
5.1 Section 2	6
5.1.1 Section 3	6
5.2 Sections Spéciales	6
5.2.1 Titres et auteurs	6
5.2.2 TOC	7
5.2.3 Images	7
6 Modes Tab	9
6.1 Listes	9
6.1.1 Césure manuelle	10
6.1.2 Enumeration	10
6.1.3 Puces	11
6.1.4 Définitions	11
6.1.5 Liste mixte	11
6.2 Autres modes Tab	12
6.2.1 Citation	12
6.2.2 Environnement Verbatim	13
7 Présentation	14
7.1 Paragraphes	14
7.2 Sauts de page	14
7.3 Lignes centrées	14
7.4 Lignes de séparation	15
7.5 Polices	15
7.5.1 Gras	16
7.5.2 Italique	16
7.5.3 Teletype et petits caractères	16
8 Cibles et Liens	17
8.1 Cibles	17
8.2 Liens	17
8.3 Adresses Web	18
8.4 Autre moyen d'éditer des cibles (Obsolète)	18
8.5 Autre moyen d'éditer des liens (Obsolète)	19
8.6 Adresses Web (Obsolète)	19
9 Divers	20
9.1 Notes de bas de page (ou notes de fin)	20
9.2 Tableaux	20
9.2.1 Lignes étendues	21

9.3 Commentaires	22
9.4 Pragmas	22
9.4.1 PASS	22
9.4.2 Variables	22

10 Remerciements	23
-------------------------	-----------

1 Introduction

Ce document est le manuel de référence pour AFT. Il n'est pas censé servir de tutoriel².

AFT est *presque* un langage de formatage de textes saisis à l'aide d'un éditeur ou d'un traitement de texte supportant les tabulations ou *espaces solides* (les espaces dont le repérage se fait sur les colonnes et dont le nombre n'augmente ni ne diminue lors du formatage). En utilisant AFT, vous ne serez plus limité à un type de fichier texte (comme avec Microsoft Word), et vous n'aurez plus à entrer la pléthore de commandes syntaxiques obscures des langages balisés (comme avec L^AT_EX ou HTML)³.

Contrairement aux autres langages balisés, AFT est conçu pour analyser et reconnaître des *modèles* plutôt que des commandes. C'est pourquoi il n'y a pas d'échappatoire ou de séquence de commande qui dit à AFT ce qu'il doit faire. Dans cette mesure, AFT traitera absolument tout ce que vous lui donnerez à manger.

Un document AFT est aisément convertible dans des formats aussi connus que HTML, RTF et L^AT_EX. Parce qu'il y a peu de commandes, apprendre à écrire un document en AFT suscite peu d'effort. Cela ne veut cependant pas dire qu'AFT est tout puissant. Pour votre information, le document vous lisez a été conçu et écrit en utilisant AFT.

1.1 Comment lire ce document

Dans ce document, le texte entré par l'utilisateur est rendu en `teletype`. Partout où vous verrez la séquence `[tab]`, cela signifiera simplement que l'utilisateur devra presser sur la touche *Tab* (ou entrer 8 espaces consécutifs). Référez vous à la section Modes Tab (§ 6) pour les instructions relatives au rapport entre les tabulations et les espaces consécutifs.

Chaque commande introduite dans ce manuel comportera une description et sera suivie par un ou plusieurs exemples de la manière dont la commande sera entrée. Dans la plupart des cas, le résultat de la commande sera affiché.

2 Lancer AFT

Pour lancer AFT, tapez simplement

```
aft NOM_DE_VOTRE_FICHER.aft
```

²Ce n'est pas tout à fait exact. Si vous le lisez de manière linéaire, cela ressemble à un tutoriel!

³L^AT_EX et HTML sont les langages balisés les plus couramment supportés; il y a moins de support pour les autres formats.

Remplacez `NOM_DE_VOTRE_FICHER` par le chemin complet de votre fichier. Il peut se terminer en `.aft` bien que ce ne soit pas de règle. Vous pouvez de même donner plusieurs noms de fichier. Par défaut, AFT se réfère au premier nom de fichier fourni pour donner le nom du fichier de sortie. Le nom du fichier de sortie comportera une extension qui reflète le type de sortie désigné par défaut. Ce réglage par défaut est choisi lors de la première installation d'AFT.

Vous pouvez ajouter quelques paramètres supplémentaires à AFT. Voici ce qu'AFT affichera si vous le lancez sans autre argument:

```
aft [--verbose] [--autonumber] [--output=file | --output=-] [--type=output_type] infile ..
```

Tout ce qui est compris entre `[]` est optionnel. Lorsque vous voyez une barre (pipe) `|` cela indique que vous pouvez renseigner l'un des deux paramètres signalé de chaque côté.

Voici une description de chaque paramètre :

-verbose Affiche les commentaires. Par défaut, AFT se contente de traiter silencieusement les fichiers (sauf en cas d'erreur). Utiliser cette fonction revient à demander à AFT de vous tenir informé de ce qu'il fait.

-autonumber Cette fonction demande à AFT de numéroter automatiquement vos sections.

-output=file | -output=- Ce paramètre demande à AFT d'envoyer le résultat de son traitement à un emplacement donné. Vous pouvez renseigner un nom de fichier (file) ou renseigner `-` qui demande à AFT d'envoyer son résultat à votre sortie standard (l'affichage écran ou `stdout`).

-type=output_type Ce paramètre demande à AFT quel type de sortie produire. Par exemple une sortie HTML est obtenue en renseignant `html`, une sortie DocBook est obtenue en renseignant `docbook`. Comme effet secondaire, cela va aussi spécifier l'extension du nom de fichier pour le fichier de sortie.

infile .. Chemin d'un ou plusieurs documents AFT devant être traité.

Il est très peu probable qu'AFT affiche des erreurs. Dans la mesure où il s'agit d'un analyseur de texte *libre en principe*, rien dans le document source ne sera jamais syntaxiquement incorrect. Dans la plupart des cas, AFT devrait produire un rendu parfait. Mais souvenez vous: *déchets à l'entrée, déchets à la sortie*⁴.

3 AFT Text: Un exemple

Le texte suivant représente une sortie exemplaire de l'analyseur AFT:

```
*Title: Catégorisation de genres musicaux
*Author: Alfred Theodore Franti
```

```
[tab]# Je suis optimiste. Jusqu'ici, la musique va
[tab] plutôt bien,du moins, elle est en bonne forme pour
```

⁴NdT - Il s'agit de l'expression 'Garbage in, garbage out', signifiant que des mauvaises données au départ ne peuvent impliquer que de mauvais résultats à la fin.

[tab] l'avenir malgré le fait qu'elle pourrait bien
 [tab] crier, un jour, comme si elle était prise en otage. --- Duke Ellington

On peut choisir plusieurs ''saveurs'' dans la musique. En effet, la musique, c'est comme la nourriture: différents goût pour différentes gens.

Mais quelles sont ces ''saveurs'' que la musique nous propose?

Voici une courte liste non exhaustive:

[tab]* Jazz
 [tab]* Classical
 [tab]* Blues
 [tab]* Country
 [tab]* Folk
 [tab]* Rock
 [tab]* Rap
 [tab]* R&B
 [tab]* Soul

Il y a bien d'autres catégories et sous-catégories dont je n'ai pas fait mention.

Après avoir passé le texte à la moulinette AFT, nous obtenons:

Musical Categorization

Alfred Theodore Franti

Je suis optimiste. Jusqu'ici, la musique va plutôt bien, du moins, elle est en bonne forme pour l'avenir malgré le fait qu'elle pourrait bien crier, un jour, comme si elle était prise en otage. — Duke Ellington

On peut choisir plusieurs *saveurs* dans la musique. En effet, la musique, c'est comme la nourriture: différents goût pour différentes gens. Mais quelles sont ces *saveurs* que la musique nous propose?

Voici une courte liste non exhaustive:

- Jazz
- Classical
- Blues
- Country
- Folk
- Rock
- Rap
- R&B
- Soul

Il y a bien d'autres catégories et sous-catégories dont je n'ai pas fait mention.

4 Revue de la syntaxe

AFT traite votre fichier d'entrée une ligne à la fois. Une ligne est un fragment de texte terminé par une nouvelle ligne, un retour chariot ou une combinaison des deux, ce qui est géré nativement par votre système d'exploitation (ne vous inquiétez pas, votre éditeur de texte gère tout cela pour vous en coulisse).

4.1 Continuité des lignes

Si le dernier caractère de votre ligne est un backslash `\`, alors la prochaine ligne est comprise comme la continuation de votre ligne courante, et toutes les choses AFT-ables qui sont effectuées ligne après ligne seront alors appliquées à toutes les lignes que vous avez déclarées continues avec un backslash.

C'est très utile lorsque vous traitez avec AFT un texte qui été *enveloppé* par votre éditeur de texte (parce que les lignes étaient trop longues) :

Voici une longue URL tapée sur 2 lignes: `http://www.maplefish\`
`.com/todd/aft.html`

Cela donne:

Voici une longue URL tapée sur 2 lignes: <http://www.maplefish.com/todd/aft.html>

Après qu'une ligne pleine (y compris celles déclarées continues) ai été analysée, AFT détermine si un mode spécial s'applique à la ligne. Les modes d'AFT sont:

1. Paragraph - le mode normal.
2. List - un mode tabulation (mode Tab).
3. Verbatim - un mode tabulation (mode Tab).
4. BlockVerbatim - un mode spécial entre `^<<` et `^>>`.
5. Quote - un mode tabulation (mode Tab).
6. Centered - un mode tabulation (mode Tab).
7. Table - un mode tabulation (mode Tab).

4.2 Les modes

Les sections sont traitées comme des modes *ponctuels*. Après un changement de section, vous entrez de nouveau un mode paragraphe. Les autres modes peuvent être compris comme des modes *collants* (vous entrez un mode et vous collez indéfiniment à ce mode).

Une ligne vide met fin à la plupart des modes et vous renvoie au mode paragraphe.

4.3 Caractères spéciaux

Tout ligne commençant avec un `#` signifie probablement que des Commentaires (§ 9.3), des Variables (§ 9.4.2) ou des Pragmas (§ 9.4) vont suivre. Une ligne commençant avec `*` indique qu'une section va suivre: voir Sections (§ 5), TOC (§ 5.2.2), ou Images (§ 5.2.3).

4.4 Filtrage des lignes

AFT essaye d'appliquer des *pré-filtres* d'après le fichier de règles (`.dat` file). Dès lors il cherche et remplace les Cibles et Liens (§ 8), et il joue sur les Polices (§ 7.5). Après que la ligne ai été complètement brutalisée par AFT, des *post-filtres* sont appliqués.

5 Sections

Toutes les sections débutent par au moins 1 (maximum 7) astérisque(s) * placée(s) dans la colonne à l'extrême gauche de la ligne. Chaque astérisque représente un niveau de section. A l'exception des Sections Spéciales (§ 5.2), un espace est placé entre le dernier astérisque et le nom de la section. Le nom de la section se termine par un retour à la ligne.

Attention: *Les espaces consécutifs sont considérés comme faisant partie du nom de section*

```
* Ceci est une section de premier niveau
** Ceci est une section de second niveau
```

Ci-dessous nous pouvons voir la hiérarchie des niveaux de section:

5.1 Section 2

Dans la mesure où le HTML supporte 6 niveaux de section, nous ne devrions pas nous contenter uniquement de celui-ci.

5.1.1 Section 3

Voici le niveau 3.

Section 4 Et maintenant le niveau 4. C'est déjà bien assez bas. Vous voulez aller plus loin?

Section 5 A partir de la section 5, le corps de texte commence à être indenté. Et ce, jusqu'à la fin de la section. ⁵

1. Les listes sont aussi indentées!
2. Tout est indenté.

Section 6

Nous commençons vraiment à être bien bas dans la hiérarchie. Méfiez vous!

1. Vous devriez vraiment penser à re-structurer votre texte.
2. Avez vous vraiment besoin d'un niveau aussi bas?

Section 7

C'est le plus bas où nous puissions aller. Cela ne doit pas apparaître dans la table des matières.

⁵du niveau 5 au niveau 7, les titres de section n'apparaîtront pas dans la table des matières... Dieu merci.

5.2 Sections Spéciales

AFT a quelques noms de section réservés. Ils sont utilisés pour fournir des services spéciaux mais qui peuvent être considérés comme basés sur le système des sections. Une section spéciale peut être indiquée par un astérisque `*` suivi immédiatement par le nom de section. (NdT: par exemple, Titre et Auteur, voir section suivante)

5.2.1 Titres et auteurs

Le titre du document peut être renseigné en utilisant le nom de section `*Title:` suivi (optionnellement) par un espace et le nom du titre.

```
*Title: Manuel de référence pour Almost Free Text (AFT)
```

Procédez de la même manière pour renseigner le nom de l'auteur du document. Au lieu de `*Title:`, entrez `*Author:`

```
*Author: Todd A. Coram
```

Si `*Title:` et `*Author:` sont les deux premières lignes du document (à l'exclusion des commentaires), ils seront utilisés dans le préambule du document. C'est très utile pour des formats comme le HTML qui nécessitent de renseigner le titre du document dans l'en-tête et non dans le corps.

5.2.2 TOC

`*TOC:*`, ou Table des Matières (Table Of Contents), est utilisé pour générer automatiquement la Table des matières du document. Cette Table des matières est renseignée par les noms des Sections (§ 5).

Si vous avez besoin de collecter vos propres informations pour la Table des matières (c'est le cas pour les sorties que ne la génèrent pas automatiquement – comme le HTML), vous devrez alors lancer la commande `aft` deux fois sur votre fichier source. Le premier passage produit un fichier TOC spécial, et le second passage insère ce fichier spécial.

```
*TOC
```

5.2.3 Images

AFT permet de procéder à un import d'image de manière très simple. La commande `*Image:` est suivie par le nom d'un fichier image qui sera intégré au document, remplaçant la ligne où la commande apparaît.

```
*Image: %aftimage%
```

La ligne ci-dessus insère le fichier `aft.gif`:



Il y a quelques variantes pour la commande `*Image:`. Vous pouvez indiquer comment l'image doit être placée dans le document avec ces commandes:

- `*Image-left:`
- `*Image-center:`
- `*Image-right:`

Les résultats sont les suivants:

`*Image-left: %aftimage%`



`*Image-center: %aftimage%`



`*Image-right: %aftimage%`



Vous pouvez aussi utiliser `*Image:` pour un rendu en cascade:

```
*Image: %aftimage%
*Image: %aftimage%
*Image: %aftimage%
```



6 Modes Tab

Lors du travail de votre document, AFT peut entrer certains *modes*. Le mode le plus courant est le *mode Tab*. Pour faire appel à ce mode, il faut entrer un caractère de tabulation (`[tab]`) en tant que premier caractère de ligne. Le mode Tab prend fin à la première ligne rencontrée qui ne commence pas par un caractère de tabulation. Cela inclu les lignes vides.

Si vous ne pouvez pas entrer de tabulation, vous pouvez utiliser des espaces. Par défaut, AFT interprétera 8 espaces consécutifs comme 1 tabulation. Vous pouvez changer ce réglage par défaut utilisant la commande :

```
#---TABSTOP=N
```

Où N est le nombre d'espaces auxquels vous voulez faire correspondre une tabulation.

Exemples:

```
#---TABSTOP=4
#---TABSTOP=8
```

6.1 Listes

Les listes sont les modes Tab les plus populaires. Un élément de liste est identifié par toute ligne contenant une tabulation initiale suivie immédiatement par l'une des séquences suivantes:

Enumeration (§ 6.1.2) Un numéro suivi de . ou).

Puces (§ 6.1.3) Une simple astérisque *.

Définitions (§ 6.1.4) Un crochet gauche [suivi de texte et finissant par un crochet droit].

Liste mixte (§ 6.1.5) Un ou plusieurs caractère(s) de tabulation(s).

Vous pouvez combiner les types de listes. Cependant, chaque fois que vous changez de type, vous commencez en fait une nouvelle liste.

Par ailleurs, chaque élément d'une liste doit suivre immédiatement le précédent. Il ne peut y avoir de ligne vide entre les éléments.

```
[tab]1. Ceci est le premier élément.
```

```
[tab]2. Ceci N'EST PAS le second élément.
```

Cela donne:

```
1. Ceci est le premier élément.
```

```
1. Ceci N'EST PAS le second élément.
```

6.1.1 Césure manuelle

Si un élément en mode Tab ne rentre pas physiquement sur une seule ligne, il peut se poursuivre sur une nouvelle ligne dans la mesure où vous n'entrez pas l'une ou l'autre des séquences décrites plus haut. Il est cependant recommandé d'utiliser un simple espace pour conserver la cohérence de votre liste.

```
[tab]* Ceci est la première ligne physique de l'élément.
```

```
[tab] Ceci est la seconde ligne physique de l'élément.
```

```
[tab]* Ceci est un nouvel élément.
```

Cela donne:

- Ceci est la première ligne physique de l'élément. Ceci est la seconde ligne physique de l'élément.
- Ceci est un nouvel élément.

Vous pouvez aussi profiter de la syntaxe d'analyse d'AFT décrite dans la section Revue de la syntaxe (§ 4) et procéder comme suit, en utilisant un backslash:

```
[tab]* Ceci est la première ligne physique de l'élément.\
Ceci est la seconde ligne physique de l'élément.
[tab]* Ceci est un nouvel élément.
```

Cela donne:

- Ceci est la première ligne physique de l'élément.Ceci est la seconde ligne physique de l'élément.
- Ceci est un nouvel élément.

Mais c'est un peu gênant...

6.1.2 Enumeration

Une liste énumérée renseigne chaque ligne par un numéro suivi de . ou). Le numéro que vous indiquez est ignoré. Une liste commence toujours l'énumération par le numéro 1. Si vous êtes dérangé par les numéros (notamment parce que AFT les ignore), vous pouvez remplacer les numéros par # (i.e. #) ou #.).

```
1. Premier élément en premier.
3. Troisième élément - ha, non! le second.
#) Et hop, je prends le bon numéro.
```

Cela donne:

1. Premier élément en premier.
2. Troisième élément - ha, non! le second.
3. Et hop, je prends le bon numéro.

Si en revanche vous voulez contrôler la numérotation, utilisez plutôt les listes de Définitions (§ 6.1.4) en renseignant vos numéros en tant que noms.

6.1.3 Puces

Une liste à puces est obtenue avec un simple astérisque *.

```
[tab]* Il y a plein d'étoiles dans le ciel.
[tab]* Les baleines ont la tête dans les nuages.
```

Cela donne:

- Il y a plein d'étoiles dans le ciel.
- Les baleines ont la tête dans les nuages.

6.1.4 Définitions

Un élément nommé s'écrit entre entre crochets. Le texte suivant cet indicateur est supplémentaire et non obligatoire.

```
[tab][Tamanoir] Mammifère friand de fourmis.
[tab][Fourmi] Une petite créature, dessert préféré des tamanoirs.
```

Cela donne:

Tamanoir Mammifère friand de fourmis.

Fourmi Une petite créature, dessert préféré des tamanoirs.

6.1.5 Liste mixte

Les listes peuvent se mélanger. Le mode mixte est automatiquement géré par AFT. Vous pouvez librement emprunter et quitter autant de niveaux que vous voulez. Une liste mixte est indiquée par des tabulations additionnelles suivies par un indicateur de liste. Les mêmes règles de Césure manuelle (§ 6.1.1) s'appliquent.

```
[tab] * Liste au niveau 1.
[tab] * Un autre objet au niveau 1
[tab][tab] et la suite de la ligne.
[tab][tab] 1. un objet niveau 2.
[tab][tab] 2. un autre objet niveau 2.
[tab][tab][tab] * Un élément niveau 3.
[tab][tab] 3. Le troisième objet de niveau 2.
[tab][tab][tab] * Un élément spécial du niveau 3.
[tab][tab][tab] * Encore un élément spécial du niveau 3.
[tab] * Retour au niveau 1.
[tab][tab] * Une nouvelle liste niveau 2.
[tab][tab][tab] 1. Un nouveau niveau 3.
[tab] 1. Un nouveau niveau 1.
[tab] 2. Un autre niveau 1.
```

Cela donne:

- Liste au niveau 1.
- Un autre objet au niveau 1 et la suite de la ligne.
 1. un objet niveau 2.
 2. un autre objet niveau 2.
 - Un élément niveau 3.
 3. Le troisième objet de niveau 2.
 - Un élément spécial du niveau 3.
 - Encore un élément spécial du niveau 3.

- Retour au niveau 1.
 - Une nouvelle liste niveau 2.
 1. Un nouveau niveau 3.
- 1. Un nouveau niveau 1.
- 2. Un autre niveau 1.

6.2 Autres modes Tab

Outre les Listes (§ 6.1), il y a d'autres modes disponibles.

6.2.1 Citation

Vous désirez entrer quelques vers ou une citation? C'est très facile. Il suffit de l'indiquer avec une tabulation suivie immédiatement de #. Toutes les lignes suivantes sont alors considérées comme faisant partie de la citation.

```
[tab]# C'est le p'tit bout d'la queue du chat
[tab]Qui passait par là. - Les Frères Jacques
```

Cela donne:

C'est le p'tit bout d'la queue du chat Qui passait par là. — Les Frères Jacques

6.2.2 Environnement Verbatim

Il s'agit des entrées en mode Tab qui ne sont ni des Listes (§ 6.1), ni des Lignes centrées (§ 7.3), ni des Tableaux (§ 9.2) ni des Citations (§ ??). A la sortie, le texte sera rendu par une police `teletype` avec un traitement limité. Les retours à la ligne sont toujours respectés dans ce mode.

Il y a deux manières d'envisager ce mode: Textuel avec tabulations (§ 6.2.2) et Textuel strict (§ 6.2.2). Chacune permet de faire apparaître le texte tel qu'il a été écrit.

Textuel avec tabulations Le mode Textuel avec tabulations est entré selon les règles des Modes Tab (§ 6) et se limite aux séquences ne relevant pas des Listes (§ 6.1).

```
[tab] 1. Cette ligne est entrée en mode tabulé et non en mode liste
[tab]   car elle commence par un espace.
[tab]
[tab]Le mode textuel tabulé conserve
[tab][tab]les retours chariot, les espaces,
[tab][tab]les tabulations, et autres maltraitements textuelles.
```

Cela donne:

1. Cette ligne est entrée en mode tabulé et non en mode liste car elle commence par un espace.

Le mode textuel tabulé conserve
 les retours chariot, les espaces,
 les tabulations, et autres maltraitements textuelles.

Textuel strict Il s'agit d'une fonction très spéciale de l'environnement verbatim. Ce n'est pas un mode Tab au sens strict, mais il vous permet d'oublier toutes les tabulations initiales que vous devez entrer en mode Tab. Le mode textuel strict commence avec la séquence `^<<` comme premier caractère de ligne, et se termine avec la séquence `^>>`.

```
^<<
while (<RF>) {
  chop;
  $lcnt++;           # increment the line count
  /^(\s*\#|\Z)/ && next; # Skip comment lines
  doStuff();
}
^>>
```

Cela donne:

```
while (<RF>) {
  chop;
  $lcnt++;           # increment the line count
  /^(\s*\#|\Z)/ && next; # Skip comment lines
  doStuff();
}
```

L'ajout d'un modificateur à la séquence du mode textuel strict permet de *filtrer* le text initial. Cela signifie que la présentation du texte est maintenue dans ce mode, mais que nous pouvons appliquer des commandes comme un changement de police. Pour entrer la commande de filtre, il suffit de taper `Filter` ou `filter` immédiatement après `^<<`.

```
^<<Filter
_while_ (<RF>) {
  _chop_;
  $lcnt++;           ''# increment the line count''
  /^(\s*\#|\Z)/ && _next_; ''# Skip comment lines''
  doStuff();
}
^>>
```

Cela donne:

```
while ($<RF$>$) {
```

```

chop;
$!cnt++;           # increment the line count
/^(\\s*\\#|\\Z)/ && next; # Skip comment lines
doStuff();
}

```

7 Présentation

AFT est très simple d'utilisation, mais la contrepartie est qu'il ne permet que peu de contrôle de la part de l'utilisateur sur le document produit. AFT n'est pas destiné à produire des dépliants, des prospectus ou des documents nécessitant une mise en page complexe.

Cela dit, il y a quelques subtilités pour vous aider à garder un certain niveau de contrôle sur la mise en page de votre document.

7.1 Paragraphes

Tous les paragraphes se terminent avec une simple ligne vide.

7.2 Sauts de page

La combinaison traditionnelle `^L` (Control L) peut être utilisée pour obtenir des sauts de page.

7.3 Lignes centrées

Une ligne de texte peut être centrée en modifiant légèrement les règles des Modes Tab (§ 6). Si vous n'êtes pas au milieu de Listes (§ 6.1), de Tableaux (§ 9.2), de Citations (§ ??) ou d'un Environnement Verbatim (§ 6.2.2), alors vous pouvez entrer une *ligne centrée*.

Une ligne centrée est une ligne de texte précédée d'au moins deux tabulations.

```

[tab][tab][tab]Salut, je suis le centre de l'univers.
[tab][tab]Je suis parfaitement autosatisfait de moi-même aussi.
[tab] N'oubliez pas.
[tab][tab]Vous ne pouvez rien centrer si vous n'êtes pas en mode Tab.

```

Cela donne:

Salut, je suis le centre de l'univers.

Je suis parfaitement autosatisfait de moi-même aussi.

N'oubliez pas.

Vous ne pouvez rien centrer si vous n'êtes pas en mode Tab.

7.4 Lignes de séparation

Une ligne horizontale simple est obtenue en tapant au moins 4 tirets sur une ligne. Vous pouvez ajouter une ligne vide avant et après la ligne de manière à insérer des retours à la ligne.

Ce qui précède

Ne doit jamais se trouver en dessous.

Cela donne:

Ce qui précède

Ne doit jamais se trouver en dessous.

7.5 Polices

AFT ne vous permet pas de changer vos polices de caractères. Cela regarde la configuration de votre navigateur web (pour le HTML) ou votre installation de T_EX (pour L^AT_EX). Cependant, AFT vous permet de changer la *tête* de votre police de caractère.

La reconnaissance typographique d'AFT procède ligne par ligne. De manière à changer l'aspect d'une partie d'un texte, vous devez l'entourer de séquences de commande de police. Le début et la fin des ces séquences doit apparaître **sur la même ligne physique**.

Il a été très `_direct et audacieux_` dans son discours.

Il a été `_très direct et
audacieux_` dans son discours.

Cela donne:

Il a été très **direct et audacieux** dans son discours. Il a été `_très direct et audacieux_` dans son discours.

Une exception à cette règle se produit lorsque vous voulez transformer un paragraphe (ou plusieurs lignes) d'un texte. Dans ce cas, vous placez le début de la séquence de commande en premier caractère dans la première ligne, et la fin de la séquence de commande en dernier caractère dans la dernière ligne.

`_Attention. Ne touchez pas à ceci. Si vous le faites cela pourrait causer
une grande catastrophe et peut-être la fin du monde.
Vous aurez été prévenu._`

Cela donne:

Attention. Ne touchez pas à ceci. Si vous le faites cela pourrait causer une grande catastrophe et peut-être la fin du monde. Vous aurez été prévenu.

7.5.1 Gras

Un texte en gras est entouré par le caractère `_`. Si vous souhaitez écrire ce caractère littéralement dans votre texte, vous devez alors effectuer un saut de ligne de manière à ce qu'il apparaisse, ou bien doubler le caractère `_`.

```
_La choucroute et la saucisse._ La__choucroute__et__la__saucisse.
```

Cela donne:

La choucroute et la saucisse. La_choucroute_et_la_saucisse.

7.5.2 Italique

Un texte en italique est entourée de la séquence de caractères `■`. Si vous désirez écrire cette séquence littéralement dans votre texte, vous devez alors effectuer un saut de ligne de manière à ce qu'il apparaisse, ou bien doubler la séquence `■`.

```
''C'était très drôle''. ''''Ce n'était pas drôle du tout''''.
```

Cela donne:

C'était très drôle. "Ce n'était pas drôle du tout".

7.5.3 Teletype et petits caractères

Un texte en Teletype est entouré par le caractère `|`. Si vous désirez écrire cette séquence littéralement dans votre texte, vous devez alors effectuer un saut de ligne de manière à ce qu'il apparaisse, ou bien doubler le caractère `|` ou bien encore faire précéder le *pipe* d'un backslash `\`. Les petits caractères s'obtiennent de la même manière, sous les mêmes conditions, en utilisant `~`.

```
|int small__number = 1 \| 2;| ~# this holds a small number \~ (1 or 2).~
```

Cela donne:

```
int small_number = 1 | 2; # this holds a small number ~ (1 or 2).
```

8 Cibles et Liens

Le format Hypertexte est supporté pour la production HTML et les Références sont supportées pour la production L^AT_EX. Les deux utilisent les mêmes séquences de commandes.

8.1 Cibles

Les cibles sont des points d'ancrage dans un document, souvent référencés par des Liens (§ 8.2).

Un cible prend la forme suivante:

```
=[texte]=
```

ou

```
=[(texte)]=
```

La différence étant que la première forme *insère* le texte à l'emplacement de la cible, alors que la seconde forme (avec les parenthèses) fournit un point de référence sans insérer de texte.

Le `texte` est utilisé par des Liens (§ 8.2).

Voici deux exemples:

=`[Votre chien mord t'il?]= Non, il ne mord pas.`

=`[(Oh oui! mord moi!)] = Ca... ce n'est pas mon chien.`

Cela donne:

Votre chien mord t'il? Non, il ne mord pas.

Ca... ce n'est pas mon chien.

8.2 Liens

Un lien peut prendre l'une de ces trois formes:

Forme #1 `[texte]`

Forme #2 `[texte (reference_texte)]`

Forme #3 `[texte (url:reference_texte)]` ou `[texte (:reference_texte)]`

Où `texte` lie la cible (§ 8.1), sauf si un `reference_texte` spécifique est précisé (et dans ce cas, `reference_texte` se réfère à la cible (§ 8.1)).

La première forme (#1) est la plus simple. Elle se réfère à une cible dans le même document. La seconde forme (#2) permet d'utiliser un texte arbitraire. La troisième forme (#3) permet d'utiliser un texte arbitraire pour se référer à une URL externe au document. `url` peut être `ftp`, `http`, `https`, `mailto` (laisser les ':'). Si vous désirez intégrer des URLs directement dans votre document, reportez vous à la section Adresses Web (§ 8.3)).

Ici, nous référons aux cibles indiquées dans la section Cibles (§ 8.1):

`[Demander a propos du chien (Votre chien mord t'il?)]!`

`[Oh oui! mord moi!]`

Voir `[Comment éviter les morsures de chien (http://www.hsus.org/ace/11858)]`.

Cela donne:

Demander a propos du chien (§ 8.1)!

Oh oui! mord moi! (§ 8.1)

Voir Comment éviter les morsures de chien <http://www.hsus.org/ace/11858>.

Cependant, qu'en est-il si vous désirez inclure un texte avec des crochet sans qu'il soit considéré comme un lien? Pour cela vous devez écrire votre texte entre crochets sur deux lignes:

```
[Coram97
]
```

Mais cela introduit un espace non désiré après 97 (i.e. `[Coram97]`).

donc vous pouvez probablement désaffecter les crochets avec un backslash `\` :

```
\[Ceci n'est pas un lien et ne comporte pas d'espace non désiré]
```

En effet, `[Ceci n'est pas un lien et ne comporte pas d'espace non désiré]`.

8.3 Adresses Web

Si vous désirez entrer des adresses web complètes, vous n'avez pas besoin d'entrer de balises spéciales. Contentez vous d'écrire l'adresse. Mais AFT n'est pas capable de reconnaître des URL sophistiquées. Vous devez rester simple (i.e. les URL contenant des paramètres http doivent être évitées).

Par exemple:

- * `http://www.maplefish.com/todd` est ma page personnelle.
- * `(http://www.maplefish.com/todd)`

cela donne:

- <http://www.maplefish.com/todd> est ma page personnelle.
- (<http://www.maplefish.com/todd>)

8.4 Autre moyen d'éditer des cibles (Obsolète)

(NdT: depuis la Version 5.08b (1avr2003) Todd A. Coram a amélioré le support AFT des cibles et liens. Obsolète est un grand mot: l'ancienne méthode (ci-dessous) n'est pas abandonnée, mais il est préférable d'utiliser la plus récente (ci-dessus)).

Un cible débute par une accolade droite } suivie par un indicateur de visibilité, et se termine par une accolade gauche {.

Il y a deux indicateurs de visibilité possibles:

1. Pour rendre la cible visible +. (le texte de la cible apparaît dans le document final)
2. Pour rendre la cible invisible -. (le texte de la cible n'apparaît pas dans le document final)

```
}-Information importante-{ Mon chien peut mordre.
}+Information très importante+{: Mon chien mord.
```

Mon chien peut mordre. Information très importante: Mon chien mord.

Il est conseillé d'utiliser des textes ne contenant pas de balises AFT ou des caractères spéciaux. Faites court et clair.

8.5 Autre moyen d'éditer des liens (Obsolète)

(NdT: même remarque que précédemment).

Un lien débute avec une accolade gauche {, suivie par un indicateur de visibilité, du texte, un texte de cible optionnel, et se termine par une accolade droite }.

Il y a deux indicateurs de visibilité possibles:

1. Pour rendre l'adresse entièrement visible +.
2. Pour rendre l'adresse à moitié visible -.

La première méthode est utilisée si le texte du lien **est** le lien lui-même. La seconde méthode est utilisée avec le caractère @ pour indiquer que la cible sera renseignée mais ne sera pas visible telle quelle. La forme est la suivante: `texte visible@texte de la cible invisible`

Les `{+Informations importantes+}` méritent d’être lues.
 Les `{-Informations@Information très importante-}` sont disponibles.

Les Informations importantes (§ ??) méritent d’être lues. Les Informations (§ 8.4) sont disponibles.

8.6 Adresses Web (Obsolète)

(NdT: même remarque que précédemment).

Les URLs peuvent être indiquées de différentes manières.

1. Adresses web. Utilisez `http:`, `file:`, `ftp:` or `mailto:` dans le nom de la cible.
2. Fichiers locaux. Utilisez `texte@:nom_du_fichier` ou `texte@:nom_du_fichier#cible`.

Voici quelques exemples:

```
* {+http://www.maplefish.com/todd+}
* {-AFT Home Page@http://www.maplefish.com/todd-}
* {-AFT fichier source du manuel@:aft-refman.aft-}
* {-AFT informations sur les commandes hypertextuelles@:aft-refman.html#Liens-}
```

Cela donne:

- <http://www.maplefish.com/todd>
- AFT Home Page <http://www.maplefish.com/todd>
- AFT fichier source du manuel [aft-refman.aft](#)
- AFT informations sur les commandes hypertextuelles [aft-refman.html#Liens](#)

9 Divers

9.1 Notes de bas de page (ou notes de fin)

AFT propose un système de notes de bas de page très simple (pour les systèmes comme le HTML, où les vraies notes de bas de page sont hors de propos, les notes seront traités comme des notes de fin).

La syntaxe des notes est une adaptation du système de Liens (§ 8.2). Vous indiquez une note en commençant la référence par `[Note:]`. L’ensemble du texte suivant `Note:` jusqu’au `’` est interprété comme une note de bas de page. Par exemple:

```
La syntaxe des notes est une adaptation du système de Liens.[Note: Ceci \
est une note servant d'exemple. Vous pouvez revenir à \
votre lecture du manuel.] Compris?
```

Cela donne:

La syntaxe des notes est une adaptation du système de Liens.⁶ Compris?

Dans l’exemple précédent, nous montrons aussi comment les longues notes (la plupart s’étendent sur plusieurs lignes) peuvent utiliser une Césure manuelle (§ 6.1.1).

⁶Ceci est une note servant d’exemple. Vous pouvez revenir a votre lecture du manuel.

9.2 Tableaux

Il y a une manière très simple de faire des tableaux vec AFT. On entre dans le mode Tableau avec une tabulation suivie d'un !. Toute ligne ne commençant plus par cette séquence met fin au tableau.

Un tableau se construit avec les éléments suivants, et dans cet ordre:

1. Légende.
2. En-têtes.
3. Données.

Les éléments de l'en-tête et les données sont séparés entre eux par un simple !. Chaque entrée (en-tête et données du tableau) comportant moins de 2 éléments (caractères) génère une alerte et se fait ignorer.

Vous pouvez utiliser !——! pour séparer les lignes du tableau (Cela favorise aussi la lisibilité du document source).

Par exemple:

```
[tab]! Dates _très_ importantes!
[tab]!-----!
[tab]! Année ! Mois ! Jour      !
[tab]!-----!
[tab]! 1966 ! Oct   ! 9         !
[tab]! 1999 ! Dec   ! 31        !
[tab]! 2000 ! Jan   ! 1         !
```

Cela donne:

Dates très importantes		
Année	Mois	Jour
1966	Oct	9
1999	Dec	31
2000	Jan	1

9.2.1 Lignes étendues

En utilisant le mode Tableau par défaut, tenter d'adapter un long texte sur une seule ligne peut s'avérer particulièrement laid. Une petite amélioration dans la présentation peut être obtenue en introduisant le pragma `#--SET-CONTROL tableparser=new`. Lorsque ce pragma est effectif, vous **devez** utiliser les séparateurs de ligne `---`. C'est ce qui permettra à AFT de détecter qu'une ligne du tableau est finie.

(NdT: le résultat semble être meilleur au format \LaTeX avec cette méthode)

Ainsi, vous pouvez produire quelque chose de ce genre:

```
#---SET-CONTROL tableparser=new
```

```
! Dates _très_ importantes          !
!-----!
! Année ! Mois ! Jour                !
!-----!
! 1966 ! Oct  ! 9 (Anniversaire       !
!      !      ! de Todd.                          !
!      !      ! Préparez les cadeaux)!
!-----!
! 1999 ! Dec  ! 31                          !
!-----!
! 2000 ! Jan  ! 1                          !
!-----!
```

Cela donne:

Dates très importantes

Année	Mois	Jour
1966	Oct	9 (Anniversaire de Todd. Préparez les cadeaux)
1999	Dec	31
2000	Jan	1

9.3 Commentaires

Si vous désirez saisir du texte qui ne doit pas être traité par AFT, il vous suffit de préfacier chaque ligne dont vous voulez qu'elle soit ignorée avec un `C` ou un `#` suivi d'au moins 3 tirets -.

```
#----- Cette ligne est un commentaire.
Maroc Ddot est
C----- Todd Coram! Nous avons percé le secret!
Le Corbeau.
```

Cela donne:

Maroc Ddot est Le Corbeau.

9.4 Pragmas

Ah! les pragmas⁷. Pour ceux qui ne sont pas entièrement satisfaits de la manipulation de texte avec AFT, vous pouvez explorer ses sombres recoins avec les pragmas suivants.

9.4.1 PASS

Si AFT ne supporte pas une caractéristique du format de sortie voulu, vous pouvez tricher et intégrer un balisage qui passera directement en sortie. Cela est possible en utilisant la commande `#-PASS-xxx`, où `xxx` est l'identité ID du format de sortie. Cette identité ID peut être trouvée dans le fichier *dat* de AFT, qui est utilisé pour décrire le format de sortie.

⁷Un pragma est une instruction spéciale adressée au compilateur pour qu'il compile d'une certaine manière

Par exemple, Le PASS suivant permet une sortie HTML brute *si et seulement si* le format de sortie indiqué est bien le HTML:

```
Est-ce bien la couleur
#---PASS-TROFF // Le langage de formatage de texte Troff
#---est trop délirant pour être supporté pour l'instant par AFT.
#---PASS-HTML <font color=red size="+1">
rouge
#---PASS-HTML </font>
?
```

Est-ce bien la couleur rouge ?

9.4.2 Variables

`#--SET` est utilisé pour assigner un bloc de texte (y compris les balises AFT) à une variable qui peut être librement invoquée et développée au long du document. (Si vous êtes un hacker pour AFT, et que vous avez jeté un oeil dans le fichier *.dat, vous avez pu voir que `#--SET` est utilisé aussi ici).

La forme de la séquence SET est:

```
#---SET variable=texte
```

Tout au long de votre document, `variable` peut être invoquée par `%variable%` et sera remplacée par `texte`.

Avant laisser exploser votre joie, notez que SET ne marche seulement que pour une ligne de texte.

Voici un exemple:

```
#---SET aftimage=aft.gif
*Image-center: %aftimage%
```

L'avantage, ici, est que vous n'avez qu'à modifier `aftimage` si vous voulez vous référer un fichier image différent, et toutes les occurrences `%aftimage%` se traduiront alors par ce fichier.

Si vous trouvez que vous utilisez trop souvent `#--PASS`, vous devriez envisager une variante basée sur SET qui (comme PASS) utilise le module de nom comme extension conditionnelle. Ainsi, pour poursuivre notre exemple précédent, si le fichier image est différent selon le type de sortie que vous voulez (NdT: un .gif pour du HTML et un .jpg pour L^AT_EX, par exemple), vous pouvez faire la chose suivante:

```
#---SET-HTML aftimage=aft.gif
#---SET-LaTeX aftimage=aft.eps
*Image-center: %aftimage%
```

Il y a une subtile mais très importante différence entre SET et SET-xxx: la première est traitée durant la phase de pré-process et la seconde est traitée durant la phase de post-process. Résultat: vous pouvez inclure une balise AFT dans SET et une balise de sortie spécifique dans SET-xxx.

10 Remerciements

Merci à Ward Cunningham (NdT: inventeur du wiki), dont l'éditeur de Site Wiki-Wiki a inspiré les phases initiales du projet AFT. Merci aussi à tous ceux qui utilisent AFT et qui, au au fil des années, m'ont communiqué des retours précieux.