Lout, éditeur de rapport

- Objet : création de rapports en langage à balise
- Niveau requis :
  débutant, avisé
- Suivi :
  en-chantier
- Commentaires : L'interface d'utilisation est la console, le rendu sera affiché par un viewer pdf (evince, acroread...)
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Introduction

Comme je le signale dans ma page personnelle, j'ai eu l'occasion d'utiliser **lout** il y a de nombreuses années pour rédiger des documents techniques importants sans avoir à me plonger dans Latex.
Par une première approche du langage à balise décrite sur la page de wikipedia Lout, on peut déjà faire d'excellents rapports avec un simple éditeur de texte.
Avec un niveau de débutant, vous allez pouvoir créer des rapports automatiques depuis les données de vos systèmes et base de données.
Même les outils pour créer les documents (listings) en PDF sont présents.

Vous trouverez toujours une utilisation à ses prouesses, même (je dirais plutôt ***raison de plus***) sous OSwindows, les 2 implémentations (linux, win32) existent.

Bien-entendu, il y a toujours quelque chose qui coince... pour moi c'est la langue de Shakespeare, peut être des traducteurs pourraient se pencher sur le ~~bébé~~épapy.

Mais dans ce minituto, je ne retiens que l'aspect “Éditeur de rapport”, par un exemple d'édition de “bon de commande” depuis un logiciel de comptabilité.

Installation

Sous Debian, depuis une console installer le package lout

```
apt-get install lout
```

Vous retrouverez l'ensemble de la documentation dans /usr/share/lout au format ps

- slide.ps - Guide initial à l'utilisation de lout
- user.ps - Documentation utilisateur
- expert.ps - pour les experts 'ils comprendront les principes
- design.ps - concepts

De plus les manuels pour lout et prg2lout sont accessibles avec la commande man lout ou man prg2lout. Il ne vous reste plus qu'à créer un dossier, y éditer un premier fichier .lt (ou .lout) et générer votre document final avec la commande lout (voir ci-après)

Sous windows, il va falloir télécharger le fichier zip d'installation depuis SourceForge

télécharger le fichier zip d'installation depuis SourceForge.
http://sourceforge.net/projects/lout/files/lout-win32/lout-3.40-win32-20140206/lout_3_40_Win32.zip/download

créer un dossier C:\lout puis extraire le zip dans ce dossier enfin ajouter le chemin c:\lout\bin dans le PATH de votre système

Suppléments de confort

Nous complètons l'installation en configurant l'éditeur vim pour la coloration syntaxique spécifique à lout. Il faut ajouter dans le répertoire $HOME/.vim le fichier suivant

lout.vim

```
" Vim syntax file
" Language:    Lout
" Maintainer:  Christian V. J. Brüssow <cvjb@cvjb.de>
" Last Change: So 12 Feb 2012 15:15:03 CET
" Filenames:   *.lout,*.lt
" URL:         http://www.cvjb.de/comp/vim/lout.vim

" $Id: lout.vim,v 1.4 2012/02/12 15:16:17 bruessow Exp $
"
" Lout: Basser Lout document formatting system.

" Many Thanks to...
"
" 2012-02-12:
" Thilo Six <T.Six at gmx dot de> send a patch for cpoptions.
" See the discussion at
http://thread.gmane.org/gmane.editors.vim.devel/32151


" For version 5.x: Clear all syntax items
" For version 6.x: Quit when a syntax file was already loaded
if version < 600
    syntax clear
elseif exists("b:current_syntax")
    finish
endif

let s:cpo_save=&cpo
set cpo&vim

" Lout is case sensitive
syn case match

" Synchronization, I know it is a huge number, but normal texts can be
```

```
" _very_ long ;-)
syn sync lines=1000

" Characters allowed in keywords
" I don't know if 128-255 are allowed in ANS-FORHT
if version >= 600
    setlocal iskeyword=@,48-57,.,@-@,_,192-255
else
    set iskeyword=@,48-57,.,@-@,_,192-255
endif

" Some special keywords
syn keyword loutTodo contained TODO lout Lout LOUT
syn keyword loutDefine def macro

" Some big structures
syn keyword loutKeyword @Begin @End @Figure @Tab
syn keyword loutKeyword @Book @Doc @Document @Report
syn keyword loutKeyword @Introduction @Abstract @Appendix
syn keyword loutKeyword @Chapter @Section @BeginSections @EndSections

" All kind of Lout keywords
syn match loutFunction '\<@[^ \t{}]\+\>'

" Braces -- Don`t edit these lines!
syn match loutMBraces '[{}]'
syn match loutIBraces '[{}]'
syn match loutBBrace '[{}]'
syn match loutBIBraces '[{}]'
syn match loutHeads '[{}]'

" Unmatched braces.
syn match loutBraceError '}'

" End of multi-line definitions, like @Document, @Report and @Book.
syn match loutEOmlDef '^//$'

" Grouping of parameters and objects.
syn region loutObject transparent matchgroup=Delimiter start='{'
matchgroup=Delimiter end='}' contains=ALLBUT,loutBraceError

" The NULL object has a special meaning
syn keyword loutNULL {}

" Comments
syn region loutComment start='\#' end='$' contains=loutTodo

" Double quotes
syn region loutSpecial start=+"+ skip=+\\\\\|\\"+ end=+"+

" ISO-LATIN-1 characters created with @Char, or Adobe symbols
```

```
" created with @Sym
syn match loutSymbols '@\(\(Char\)\|\(Sym\)\)\s\+[A-Za-z]\+'

" Include files
syn match loutInclude '@IncludeGraphic\s\+\k\+'
syn region loutInclude
start='@\(\(SysInclude\)\|\(IncludeGraphic\)\|\(Include\)\)\s*{'
end='}'

" Tags
syn match loutTag
'@\(\(Tag\)\|\(PageMark\)\|\(PageOf\)\|\(NumberOf\)\)\s\+\k\+'
syn region loutTag start='@Tag\s*{' end='}'

" Equations
syn match loutMath '@Eq\s\+\k\+'
syn region loutMath matchgroup=loutMBraces start='@Eq\s*{'
matchgroup=loutMBraces end='}' contains=ALLBUT,loutBraceError
"
" Fonts
syn match loutItalic '@I\s\+\k\+'
syn region loutItalic matchgroup=loutIBraces start='@I\s*{'
matchgroup=loutIBraces end='}' contains=ALLBUT,loutBraceError
syn match loutBold '@B\s\+\k\+'
syn region loutBold matchgroup=loutBBraces start='@B\s*{'
matchgroup=loutBBraces end='}' contains=ALLBUT,loutBraceError
syn match loutBoldItalic '@BI\s\+\k\+'
syn region loutBoldItalic matchgroup=loutBIBraces start='@BI\s*{'
matchgroup=loutBIBraces end='}' contains=ALLBUT,loutBraceError
syn region loutHeadings matchgroup=loutHeads
start='@\(\(Title\)\|\(Caption\)\)\s*{' matchgroup=loutHeads end='}'
contains=ALLBUT,loutBraceError

" Define the default highlighting.
" For version 5.7 and earlier: only when not done already
" For version 5.8 and later: only when an item doesn't have
highlighting yet
if version >= 508 || !exists("did_lout_syn_inits")
    if version < 508
        let did_lout_syn_inits = 1
        command -nargs=+ HiLink hi link <args>
    else
        command -nargs=+ HiLink hi def link <args>
    endif

    " The default methods for highlighting. Can be overrriden later.
    HiLink loutTodo Todo
    HiLink loutDefine Define
    HiLink loutEOmlDef Define
    HiLink loutFunction Function
    HiLink loutBraceError Error
```

```
    HiLink loutNULL Special
    HiLink loutComment Comment
    HiLink loutSpecial Special
    HiLink loutSymbols Character
    HiLink loutInclude Include
    HiLink loutKeyword Keyword
    HiLink loutTag Tag
    HiLink loutMath Number

    " HiLink Not really needed here, but I think it is more consistent.
    HiLink loutMBraces loutMath
    hi loutItalic term=italic cterm=italic gui=italic
    HiLink loutIBraces loutItalic
    hi loutBold term=bold cterm=bold gui=bold
    HiLink loutBBraces loutBold
    hi loutBoldItalic term=bold,italic cterm=bold,italic
gui=bold,italic
    HiLink loutBIBraces loutBoldItalic
    hi loutHeadings term=bold cterm=bold guifg=indianred
    HiLink loutHeads loutHeadings

    delcommand HiLink
endif

let b:current_syntax = "lout"

let &cpo=s:cpo_save
unlet s:cpo_save

" vim:ts=3:sw=4:nocindent:smartindent:
```

Essais initiaux

Nous allons maintenant tester notre installation
Pour plus de simplicité nous utiliserons la ligne de commande, Ouvrir une console texte (terminal),

Créer tout d'abord un dossier de travail pour y déposer le fichier de test

```
mkdir test_lout; cd test_lout
```

Copier ce fichier d'exemple dans le dossier test_lout

essai.lout

```
@SysInclude { doc }
@Doc @Text @Begin
@Display @Heading { Démarrage rapide de Lout }
Ce Minituto de @I Lebardix sera votre premier test de la puissance de
```

```
lout 'The Lout document formatting system has been designed with the
needs of the ordinary user very
much in mind. Although the features of Lout are virtually endless, and
include mathematical
equations, diagrams made from lines and shapes, bibliographic
databases, and so on, the system
is very simple to use.'
@PP
45d @Rotate 1.5 @Scale @Box {
soit mal traduit
}
Le Système de formatage de Lout a été conçu, tout à fait dans l'esprit,
des besoins d'un utilisateur ordinaire.
Bien que les caractéristiques de Lout soient pratiquement infinies, et
comprennent mathématique
équations, schémas à base de lignes et de formes, bases de données
bibliographiques, et ainsi de suite, le système
est très simple à utiliser.
@End @Text
```

Maintenant générer le fichier pdf essai.pdf avec la commande ci-après

```
marc@nimbus:~/test_lout$ lout -PDF essai.lout  -o essai_warn.pdf
```

Sans doute obtiendrez-vous les mêmes messages d'avertissements, le fichier essai.lout à un encodage des caractères en UTF8, alors que lout attend un encodage iso-8859.

```
lout file "essai.lout":
    12,4: character "\250" replaced by space (it has no glyph in font Times
Base)
   12,55: character "\240" replaced by space (it has no glyph in font Times
Base)
   14,22: character "\240" replaced by space (it has no glyph in font Times
Base)
  14,112: character "\250" replaced by space (it has no glyph in font Times
Base)
    15,5: character "\250" replaced by space (it has no glyph in font Times
Base)
   15,18: character "\240" replaced by space (it has no glyph in font Times
Base)
```

Voici le résultat. à comparer avec le fichier essai_warn.pdf que vous trouverez dans votre dossier de travail.

Pour régler ce problème, sur l'instant nous allons convertir le fichier initial au bon encodage avec la commande iconv

```
marc@nimbus:~/test_lout$ iconv -t ISO-8859-15 essai.lout | lout  -PDF  -o
essai.pdf
```

Le fichier sera alors généré sans erreur.... et le fichier final file-R24c40b97487fcd4b3e383b219d7d244b.pdf, vous trouverez le fichier essai.pdf dans le dossier de travail

---

Bon voilà, l'essai de fonctionnement est satisfaisant, mais écrire des livres n'est pas le but recherché.

Le but de ce minituto est de créer des rapports d'état pour des documents type “bons de commande” sans autre intervention de l'utilisateur final.

Écrire un livre ou faire un rapport avec lout peut être complexe, faire un modèle sera aussi compliqué mais restera valable pour de multiples éditions tandis que pour un livre, un rapport, une présentation vous devrez reprendre la conception à chaque nouveau document.

Donc entrons dans la réalisation de modèle.

Concevoir un modèle

Utiliser le modèle

à suivre