PERL & UTF-8

Eric BERTHOMIER (eric.berthomier@free.fr)

31 août 2013

Introduction

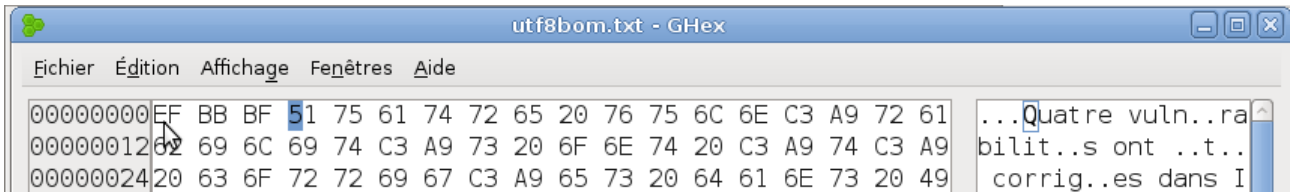
UTF-8 tout le monde connaît ou je devrai dire devrait connaître ;-D
En fait jolie utopie, nous l'utilisons ou non mais au final dès que l'on creuse un peu nombre de personnes sont comme moi ... difficile à comprendre ...
Ce document a pour but d'explorer UTF8 et Perl ...

Mon texte en UTF-8

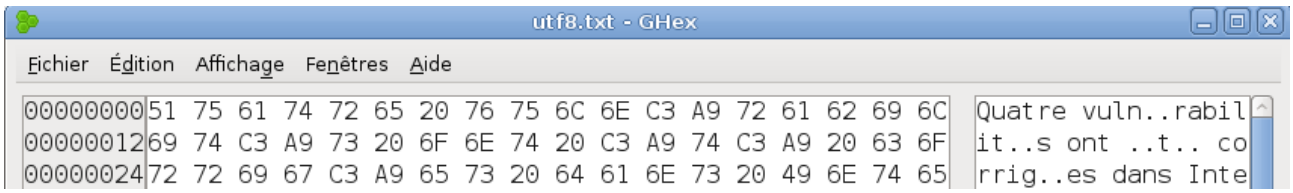
Mon éditeur favori est Scite (http://www.scintilla.org/SciTE.html), aussi lorsque je veux choisir de taper en UTF8 dans mon fichier, je fais File → Encoding … et là

2 choix :

- UTF-8 with BOM : BOM (Byte Order Mark) est comme son nom l'indique un caractère qui va permettre d'identifier que votre fichier est en UTF-8. Ce caractère est ajouté en début de fichier et va permettre d'identifier ce texte comme un fichier en UTF-8 (UTF-8 text files often start with the UTF-8 encoding of the same character, EF BB BF[1]).



- UTF-8 : notre fichier sera en UTF-8 mais aucun caractère spécial ne sera inséré.
-



Quelle incidence ?

Sur un fichier texte, pratiquement aucune tant qu'aucun accent ne vient pointer le bout de son nez … bien que la commande file nous fasse la remarque :

```
eric@souricier:~$ file utf8bom.txt
```

1 http://en.wikipedia.org/wiki/Magic_number_%28programming%29

```
utf8bom.txt: UTF-8 Unicode (with BOM) text, with no line terminators
eric@souricier:~$ file utf8.txt
utf8.txt: UTF-8 Unicode text, with no line terminators
```

Voyons maintenant sur un fichier Perl, nous éditons donc un premier fichier dépourvu d'accents (ne compliquons pas les choses) et l'enregistrons sous les noms :

- bonjour.pl (sans BOM)
- bonjourbom.pl (avec BOM)

```
#!/usr/bin/perl -w
use strict;


print "Hello World !\n";
```

Enregistré en UTF8 sans BOM, le fichier s'exécute normalement.

```
eric@souricier:~$ ./bonjour.pl
Hello World !
```

Enregistré en UTF8 avec BOM, les problèmes commencent :

```
eric@souricier:~$ ./bonjourbom.pl
./bonjourbom.pl: line 1: #!/usr/bin/perl: Aucun fichier ou dossier de ce type
./bonjourbom.pl: line 2: use : commande introuvable
Warning: unknown mime-type for "Hello World !\n" -- using "application/octet-
stream"
Error: no such file "Hello World !\n"
```

En effet le premier caractère trouvé n'est pas le shebang[2] mais le BOM (« *In unix-like OSes, BOM for UTF-8 conflicts with the unix shebang line hack* »[3]).

Une solution consiste alors à exécuter le programme par l'exécution de PERL :

```
eric@souricier:~$ perl bonjourbom.pl
Hello World !
```

Et avec les accents ?

Scite

Dans l'éditeur de texte, si je choisis l'encodage UTF-8 une fois mon texte saisi, il apparaît de jolie fioriture en lieu et place de mes accents …

Plusieurs choix,

- rechercher, remplacer,
- réécriture
- utilisation de iconv[4] dans le cas d'un gros fichier …

2 http://fr.wikipedia.org/wiki/Shebang
3 http://xahlee.info/comp/unicode_BOM_byte_orde_mark.html
4 http://eric.berthomier.free.fr/spip.php?article56

idéalement, penser à choisir l'encodage avant de commencer la frappe s'avère tout de même payant.

Perl

Reprenons notre petit programme et modifions le pour afficher des caractères accentués :

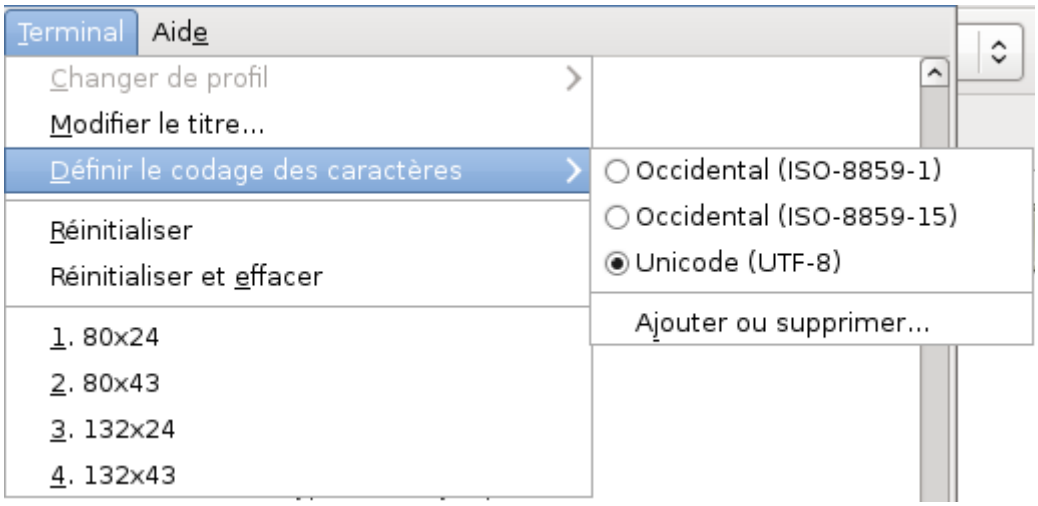
```
#!/usr/bin/perl -w
use strict;

print "dans les aquifères, c'est-à-dire la vulnérabilité ... !\n";
```

Quel que soit le programme l'ensemble des informations s'affichent correctement à mon écran :

```
eric@souricier:~$ perl bonjourbom.pl
dans les aquifères, c'est-à-dire la vulnérabilité ... !
eric@souricier:~$ ./bonjour.pl
dans les aquifères, c'est-à-dire la vulnérabilité ... !
```

Oui mais pour quelle raison ?

En fait dans la fenêtre de Terminal, il est possible de choisir l'encodage de ce qui va être affiché :



Par chance, par défaut sur ma Debian, il est en UTF-8.

Si je change cet encodage en ISO-8859-15 par exemple, je retrouve mes hiéroglyphes :

```
eric@souricier:~$ ./bonjour.pl
dans les aquifÃšres, c'est-Ã -dire la vulnÃ©rabilitÃ© ... !
```

Mon programme en ISO-8859-15 ?

Reprenons nos exemples afin de bien comprendre et codons le tout en ISO-8889-15.

```
iconv -f utf8 -t iso8859-15 bonjour.pl > bonjouriso15.pl
chmod +x bonjouriso15.pl
```

Cette fois c'est lorsque le Terminal est en UTF8 qu'apparaissent les symboles cabalistiques.

Alors, comment faire ? J'aimerai conserver mon encodage ISO-8859-15 et en même temps que ce soit propre dans ma console en UTF8.

Eh bien, il suffit de le dire au langage PERL qui une fois de plus se chargera de tout pour vous.

Ajouter la ligne :

```
binmode(STDOUT, ":utf8");
```

après le use strict ; et vérifier que votre console est bien en UTF8.

```
#!/usr/bin/perl -w
use strict;
binmode(STDOUT, ":utf8");

print "dans les aquifères, c'est-à-dire la vulnérabilité ... !\n";
```

L'exécution de votre script donnera alors de l'UTF-8 malgré le fait d'avoir codé en ISO8859-15,

UTF-8 par défaut

Cela vous est peut être arrivé si vous avez fermé bonjour.pl puis réouvert, l'encodage de ce dernier est repassé à ISO-8859-15. Pour éviter ce désagrément, il existe une astuce, il faut ajouter la ligne suivante :

```
-*- coding: utf-8 -*-
```

dans votre code[5]. Ce qui nous donne :

```
#!/usr/bin/perl -w
# -*- coding: utf-8 -*-
use strict;

binmode(STDOUT, ":utf8");
print "dans les aquifères, c'est-à-dire la vulnérabilité ... !\n";
```

5 http://eric.berthomier.free.fr/spip.php?article70