

# Fiche n° 11 – Lecture d'un fichier texte et Réalisation d'un inventaire

Ce script peut être utilisé pour plusieurs besoins ; réaliser un inventaire ou sélectionner en vue d'un traitement particulier, une série de documents à partir du code à barres ou de la référence. L'utilisation d'une douchette de lecture des codes à barres permet d'accélérer le pointage des documents concernés.

**Son principe est simple :**

- 1 - Lire un fichier texte contenant une liste de code à barres,
- 2 - Rechercher dans la table Biblio.db la présence de ces codes à barres (la recherche peut s'effectuer soit sur la rubrique reference soit sur la rubrique code à barres)
- 3 - Si le code à barres est présent dans la table on met la valeur 1 dans la rubrique Xsignet

A la fin du traitement il est possible de demander l'affichage de la liste des documents contenant un signet (Menu Signets / Extraire et imprimer les documents avec signet ou Menu Filtrer / Documents avec signet)

Ce traitement peut être effectué plusieurs fois avec des listes différentes, les signets placés au cours d'un traitement précédent sont conservés.

```
var
fbi          FileBrowserInfo
fichierSélectionné String
txt1        textstream
chain1      string
tc          tcursor
endvar
; Ce script permet de placer un signet sur tous les documents dont le code
barre est présent dans un fichier texte préalablement constitué
; Il peut être utilisé notamment pour réaliser un inventaire ou tout
simplement pour effectuer une sélection de documents

; choix du fichier contenant les codes barres
fbi.AllowableTypes = fbfiles ; cherche tous les fichiers quelle que soit
l'extension
fbi.title = "Sélection du fichier source contenant les code barre des
notices à supprimer"
fbi.path = "c:\\"
; Vous pouvez également indiquer un chemin complet :
; fbi.path = "c:\program files\Bib\Pap\Papyrus\Export\"

; affiche Parcourir et traite la sélection de l'utilisateur
if fileBrowser(fichierSélectionné, fbi) then
    fichieraimporter = fbi.drive+fbi.path+fichiersélectionné
else
return ; l'utilisateur a annulé sa demande
endif
```

```

setmouseshape(mousewait)

; ***** LECTURE DES CODE A BARRES
; ***** ouverture du fichier contenant les codes à barre
if not txt1.open(fichierainporter,"w")
then msginfo("Problème d'ouverture du fichier ",
"Le fichier n'est pas valide"
+chr(10)+"ou son chemin d'accès est trop long"
+chr(10)+"Faites un nouvel essai en copiant le fichier"
+chr(10)+"dans le répertoire export de Papyrus") return endif

; ***** DEBUT DU TRAITEMENT
finnotice = 1
numnot = 0
noticetraite = 0
tc.open("biblio.db")
tc.edit()
while finnotice = 1
numnot = numnot + 1
txt1.readline(chain1)
; utiliser la ligne suivante si votre fichier texte contient la rubrique
code barre de la table biblio
if tc.locate(33,chain1) then tc.xsignet = 1 noticetraite = noticetraite + 1
endif
; utiliser la ligne suivante si votre fichier texte contient la rubrique
reference de la table bilbio
;if tc.locate(1,chain1) then tc.xsignet = 1 noticetraite = noticetraite + 1
endif
if txt1.eof() then finnotice = 0 endif
endwhile
nontrouve = numnot - noticetraite
msginfo("Nombre de documents validés : ",strval(noticetraite))
msginfo("Nombre de documents non trouvés dans Papyrus : ",
strval(nontrouve))

```

Nous rencontrons dans ce script de nouvelles instructions :

### **FileBrowser**

Et

**Readline (utilisé avec Open et EOF)**

#### **A - Utilisation de FileBrowser**

Le code suivant appelle la fenêtre de sélection de fichier et attend que vous choisissiez un fichier. Puis il affiche les informations relatives à votre choix dans une fenêtre d'information.

```

var
  fbi FileBrowserInfo
; Déclare une variable qui utilise la structure
; de l'enregistrement FileBrowserInfo prédéfinie
selectedFile String
endVar

; Les instructions suivantes définissent les paramètres de la recherche
; Cherche dans le répertoire EXPORT
fbi.Alias = ":EXPORT:" ; autres possibilités ":BIBMEDIA :"
; Cherche les fichiers de type tables (.db)
fbi.AllowableTypes = fbTable
; Affiche la fenêtre Parcourir et traite le choix de l'utilisateur
if fileBrowser(selectedFile, fbi)
  then msginfo("Vous avez sélectionné ", selectedFile)
  else
    msginfo("Vous avez sélectionné"," Annuler")
endif

```

Au lieu de fbi.Alias on peut utiliser fbi.path en indiquant le chemin d'accès absolu

Exemple :

```
fbi.path = "c:\\"
```

L'instruction fbi.AllowableTypes = fbTable précise le type de fichier que l'on recherche.

**Liste (non exhaustive) des constantes utilisables à la place de fbTable :**

fbAllTables	Tous types de tables supportées par Paradox (*.db, *.dbf, etc.)
fbASCII	Fichiers texte (*.txt)
fbBitMap	Graphiques bitmap (*.bmp)
fbDBase	Tables dBASE (*.dbf)
fbExcel	Feuilles de calcul EXCEL (*.xls)
fbFiles	Tous fichiers (*.*)
fbForm	Fiches Paradox (*.fsl, *.fdl).
fbGraphic	Fichiers graphiques (*.bmp, *.eps, *.gif, *.pcx, *.tif)
fbLotus1	Feuilles de calcul Lotus 1-2-3 v. 1 (*.wks).
fbLotus2	Feuilles de calcul Lotus 1-2-3 v. 2 (*.wk1)
fbParadox	Tables Paradox (*.db)
fbQuattro	Feuilles de calcul Quattro (*.wkq)
fbQuattroPro	Feuilles de calcul Quattro Pro (*.wq1)
fbQuattroProWindows	Classeurs Quattro Pro pour Windows (*.wb1)
fbQuery	Fichiers de requêtes (*.qbe).
fbScript	Scripts ObjectPAL (*.ssl, *.sdl)
fbSQL (5.0)	Fichiers SQL (*.sql).
fbTable	Tous les types de tables supportés par Paradox (*.db, *.dbf, etc.).
fbTableView	Fichiers de visualisation des tables Paradox (*.tv).
fbText	Tous fichiers texte (*.txt, *.pxt, *.rtf).

## A - Utilisation de Readline

L'instruction Readline lit un fichier texte, mais au préalable il faut ouvrir le fichier texte concerné.

L'instruction **Open** est utilisée pour ouvrir un fichier texte dans le mode précisé.

### Syntaxe

Variable textstream . OPEN(Non du fichier, "mode d'ouverture")

Exemple :

```
txt1.open(fichieraimporter,"w")
```

### Description

open ouvre nomFichier dans le mode d'ouverture précisé.

Spécification de mode	Description
"a"	ajout à la fin et lecture
"r"	lecture seule
"w"	écriture et lecture
"nw"	nouveau fichier, lecture et écriture

Note Les instructions suivantes sont équivalentes :

```
ts.open("nouv.txt", "NW")  
ts.create("nouv.txt")
```

Ouvrir un fichier dans un mode autre que "a" (ajout) met le pointeur au début du fichier.

Avec un chemin d'accès DOS complet ou un alias, il est possible de préciser le répertoire dans lequel créer le fichier. Si aucun chemin d'accès ni alias n'est précisé, Le script crée le fichier dans le répertoire de travail (Bib/Pap/Papyrus).

Si la méthode réussit, elle renvoie True ; sinon, elle renvoie False.

### ReadLine

Lit une ou plusieurs lignes d'un fichier texte.

### Syntaxe

1. readLine (var valeur String ) Logical
2. readLine ( var tableauChaine Array[ ] String ) Logical

### Description

readLine lit les caractères d'une ligne de texte dans un fichier jusqu'à ce que la paire CR/LF soit rencontrée (ou que 32767 caractères soient lus), et déplace met le pointeur de fichier juste après la paire CR/LF (ou après le

32767 ème caractère). `readLine` commence la lecture à la position en vigueur du pointeur. Si cette méthode réussie, elle renvoie `True` ; sinon, elle renvoie `False`.

La syntaxe 1 enregistre une seule ligne dans `valeur`. La paire CR/LF n'est pas enregistrée.

La syntaxe 2 enregistre tout le fichier dans `tableauChaine`, où `tableauChaine` est un tableau de chaînes redimensionnable et chaque élément du tableau enregistre une ligne du fichier. La paire CR/LF n'est pas enregistrée.

### **EOF**

Pour vérifier la fin d'un fichier texte on utilise `EOF`.

### **Syntaxe**

`eof ( )` Logical

Exemple :

```
if txt1.eof() then finnotice = 0 endif
```

### Description

`eof` renvoie `True` si un déplacement tente d'envoyer le pointeur au-delà de la fin d'un fichier texte ; sinon, elle renvoie `False`.