File

COLLABORATORS			
	TITLE : File		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 23, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

File

1.1 File V1.00

```
PureBasic - File library V1.00

  'File' signifie 'fichier' en français. C'est la manière principale
  de stocker des informations sur une unité (diquette, disque dur,
  ram disque...). Cette bibliothèque va vous permettre de créer,
  ouvrir, lire des fichiers et d'y écrire des informations.


Commandes disponibles:

   CloseFile
   CreateFile
   Eof
   FileSeek
   InitFile
   Loc
   Lof
   OpenFile
   ReadByte
   ReadFile
   ReadLong
   ReadString
   ReadWord
   UseFile
   WriteByte
   WriteLong
   WriteString
   WriteWord

Example:

   File demo
```

1.2 closefile

Syntaxe

```
CloseFile(#File)
```

Description

Ferme le fichier spécifié. Vous ne pourrez plus accéder à ce fichier ultérieurement (ni en lecture, ni en écriture). Les accès aux fichiers étant tamponnées (buffered) vous devez fermer un fichier pour être sûr que toutes les informations que vous avez écrites dans ce fichier soient effectivement écrites physiquement sur le disque.

Note: A la fin d'un programme, le PureBasic ferme automatiquement tous les fichiers préalablement ouverts, donc il ne pas nécessaire que vous le fassiez explicitement.

#File: Identifiant numérique du fichier à fermer.

1.3 createfile

Syntaxe

```
Result = CreateFile(#File, FileName$)
```

Description

Crée un nouveau fichier entièrement vide. Si un fichier du même nom existait déjà, alors il est totalement effacé ! Prenez garde à cela. Si le 'Résultat' est NUL, alors la création du fichier a échouée. Vous devez tester obligatoirement ce résultat, car accéder en lecture/écriture dans un fichier qui n'existe pas cause de sevères crashs.

#File: Identifiant numérique du fichier qui sera créé.

FileName$: Nom du nouveau fichier.

1.4 eof

Syntaxe

```
Result = Eof()
```

Description

'Eof' est l'abréviation anglais de 'End Of File' c'est à dire 'Fin du fichier' en français. Cette fonction retourne la valeur NULLE tant que vous n'avez pas atteint la fin du fichier.

1.5 fileseek

Syntaxe

```
FileSeek(NewPosition)
```

Description

Change la position du pointer interne du fichier courant. Cette position est exprimée en octet.

NewPosition: valeur de la nouvelle position du pointeur interne.

1.6 initfile

Syntaxe

```
Résultat = InitFile(#NumMaxFiles)
```

Description

Initialise l'environnement nécessaire à la gestion des futurs fichiers. Vous devez appeler cette fonction avant toutes les autres fonctions de cette bibliothèque.

#NumMaxFiles: Nombre maximum de fichiers à gérer.

1.7 loc

Syntaxe

```
Position = Loc()
```

Description

Renvoie la position actuelle du pointeur interne du fichier courant.

1.8 lof

Syntaxe

```
Length = Lof()
```

Description

'Lof' est l'abréviation anglaise de 'Length Of File' c'est à dire la longueur du fichier. Cette fonction retourne la taille du fichier courant.

1.9 openfile

```
  Syntaxe

Result = OpenFile(#File, FileName$)

  Description

Ouvre le fichier spécifié, ou crée un nouveau fichier s'il n'existe pas.
Vous pouvez ensuite lire et écrire dans ce fichier. Si le 'Résultat' est
NUL alors le fichier n'a pu être ouvert. Vous devez tester obligatoirement
ce résultat, car accéder en lecture/écriture dans un fichier qui n'existe pas
cause de sevères crashs.


#File: Identifiant numérique du fichier qui sera ouvert

FileName$: Nom du fichier à ouvrir.
```

1.10 readbyte

```
  Syntaxe

Number.b = ReadByte()

  Description

Lit 1 octet dans le fichier courant, et incrémente la position du pointeur
interne de 1.
```

1.11 readfile

```
  Syntaxe

Result = ReadFile(#File, FileName$)

  Description

Ouvre un fichier existant en lecture seule (pas d'écriture possible). Si le ↩
   fichier
n'existe pas, il n'est pas créé. Si le 'Resultat' est NUL, alors le fichier n'a
pas pu être ouvert. Vous devez tester obligatoirement ce résultat, car accéder ↩
   en
lecture/écriture dans un fichier qui n'existe pas cause de sevères crashs.


#File: Identifiant numérique du fichier qui sera ouvert

FileName$: Nom du fichier à ouvrir.
```

1.12 readlong

Syntaxe

```
Number.l = ReadLong()
```

Description

Lit 4 octets dans le fichier courant, et incrémente la position du pointeur interne de 4.

1.13 readstring

Syntaxe

```
Text.s = ReadString()
```

Description

Lit une chaine de caractère dans le fichier courant. C'est l'opération inverse de la commande WriteString() .

1.14 readword

Syntaxe

```
Number.w = ReadWord()
```

Description

Lit 2 octets dans le fichier courant, et incrémente la position du pointeur interne de 2.

1.15 usefile

Syntaxe

```
UseFile(#File)
```

Description

Change le fichier courant par celui spécifié.

#File: Identifiant numérique du nouveau fichier courant.

1.16 writebyte

Syntaxe

```
WriteByte(Number)
```

Description

Ecrit un nombre de type 'Byte' (1 octet) dans le fichier courant.
Le fichier doit être accessible en écriture.

Number: nombre à écrire dans le fichier.

1.17 writelong

Syntaxe

```
WriteLong(Number)
```

Description

Ecrit un nombre de type 'Long' (4 octets) dans le fichier courant.
Le fichier doit être accessible en écriture.

Number: nombre à écrire dans le fichier.

1.18 writestring

Syntaxe

```
WriteString(Text$)
```

Description

Ecrit la chaîne de caractères donnée dans le fichier courant.
Le fichier doit être accessible en écriture.

Text$: Chaîne de caractères à écrire.

1.19 writeword

Syntaxe

```
WriteWord(Number)
```

Description

Ecrit un nombre de type 'Word' (2 octets) dans le fichier courant.
Le fichier doit être accessible en écriture.

Number: nombre à écrire dans le fichier.