

File

COLLABORATORS

	<i>TITLE :</i> File		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 19, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	File	1
1.1	File V1.00	1
1.2	closefile	1
1.3	createfile	2
1.4	eof	2
1.5	fileseek	2
1.6	initfile	3
1.7	loc	3
1.8	lof	3
1.9	openfile	3
1.10	readbyte	4
1.11	readfile	4
1.12	readlong	4
1.13	readstring	4
1.14	readword	5
1.15	usefile	5
1.16	writebyte	5
1.17	writelong	5
1.18	writestring	6
1.19	writeword	6

Chapter 1

File

1.1 File V1.00

Pure Basic - File library V1.00

Die Datei ist die gebräuchlichste Art des Speicherns auf heutigen Computern. Mit PureBasic verwalten Sie diese auf sehr einfachem und optimiertem Weg. Eine beliebige Zahl Dateien kann gleichzeitig verwaltet werden. Diese Bibliothek benutzt gebufferte Funktionen, um die Schreib- und Lesegeschwindigkeit zu erhöhen.

Befehlsübersicht in alphabetischer Reihenfolge:

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

Beispiel:

```
File demo
```

1.2 closefile

Syntax

```
CloseFile(#File)
```

Beschreibung

Schließt die angegebene Datei #File, welche damit nicht weiter benutzt werden kann. Beim Schließen einer Datei wird der Buffer effektiv auf Disk abgespeichert.

Hinweis: Am Programmende ist PureBasic "klug" genug, um alle offenen Dateien automatisch zu schließen. Deshalb müssen Sie dies nicht unbedingt selbst erledigen.

1.3 createfile

Syntax

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

Beschreibung

Öffnet eine leere Datei. Existierte die angegebene Datei bereits, wird diese geöffnet und durch eine leere ersetzt! Seien Sie vorsichtig! Ergibt 'result' nicht NULL, konnte die Datei angelegt werden und Sie erhalten den AmigaOS Zeiger auf diese Datei (für fortgeschrittene Programmierer). Ergibt das Resultat NULL, ist die Erstellung fehlgeschlagen. Dies muß immer getestet werden, da das Ausführen von Operationen auf nicht erstellte Dateien zu schlimmen Abstürzen führt.

1.4 eof

Syntax

```
Result = Eof()
```

Beschreibung

EOF steht für 'End Of File' (Ende der Datei). Die Funktion gibt true (wahr) zurück, wenn Sie das Ende der Datei erreicht haben, sonst falsch (false).

1.5 fileseek

Syntax

```
FileSeek(NewPosition)
```

Beschreibung

Ändert den Zeiger innerhalb der Datei auf die angegebene Position.

1.6 initfile

Syntax

```
Result = InitFile(#NumMaxFiles)
```

Beschreibung

Mit die dieser Funktion wird die gesamte Datei-Funktionen Umgebung zur späteren Benutzung initialisiert. Sie müssen diese Funktion vor einer ersten anderen Dateifunktion aufrufen.

#NumMaxFiles = maximale Zahl der Dateien, die Sie gleichzeitig verwalten müssen

1.7 loc

Syntax

```
Position = Loc()
```

Beschreibung

Gibt den aktuellen Zeiger (Position) innerhalb der Datei zurück.

1.8 lof

Syntax

```
Length = Lof()
```

Beschreibung

LOF steht für 'Length Of File' (Länge der Datei). Es gibt die Länge der aktuellen Datei zurück.

1.9 openfile

Syntax

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

Beschreibung

Öffnet die angegebene Datei oder erstellt eine, falls noch keine existiert. Sie können nun Lese- und Schreibfunktionen in dieser Datei ausführen. Ist 'Result' nicht NULL, wurde die Datei erfolgreich geöffnet und Sie erhalten den AmigaOS Zeiger auf die Datei (für fortgeschrittene Programmierer). Ist 'Result' gleich NULL, ist das Öffnen bzw. Erstellen der Datei fehlgeschlagen. Dies muß immer getestet werden, da das Ausführen von Operationen auf nicht erstellte Dateien zu schlimmen Abstürzen führt.

1.10 readbyte

Syntax

```
Number.b = ReadByte()
```

Beschreibung

Liest ein Byte aus der aktuellen Datei ein.

1.11 readfile

Syntax

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

Beschreibung

Öffnet eine existierende Datei ausschließlich für Lese-Operationen. Ist 'Result' nicht NULL, wurde die Datei erfolgreich geöffnet und Sie erhalten den AmigaOS Zeiger auf die Datei (für fortgeschrittene Programmierer). Ist 'Result' gleich NULL, wurde die Datei nicht gefunden oder konnte nicht geöffnet werden. Dies muß immer getestet werden, da das Ausführen von Operationen auf nicht erstellte Dateien zu schlimmen Abstürzen führt.

1.12 readlong

Syntax

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

Beschreibung

Liest einen Long-Wert (4 Bytes) aus der aktuellen Datei ein.

1.13 readstring

Syntax

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

Beschreibung

Liest einen String aus der aktuellen Datei ein.

1.14 readword

Syntax

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

Beschreibung

Liest einen Word-Wert aus der aktuellen Datei ein.

1.15 usefile

Syntax

```
UseFile(#File)
```

Beschreibung

Ändert die aktuelle Datei auf die angegebene.

1.16 writebyte

Syntax

```
WriteByte(Number)
```

Beschreibung

Schreibt einen Byte-Wert in die aktuelle Datei. Die Datei muß mit Schreib-Unterstützung geöffnet worden sein (d.h. nicht mit ReadFile()).

1.17 writelong

Syntax

```
WriteLong(Number)
```

Beschreibung

Schreibt einen Long-Wert in die aktuelle Datei. Die Datei muß mit Schreib-Unterstützung geöffnet worden sein (d.h. nicht mit `ReadFile()`).

1.18 **writestring**

Syntax

```
WriteString(Text$)
```

Beschreibung

Schreibt einen String in die aktuelle Datei. Die Datei muß mit Schreib-Unterstützung geöffnet worden sein (d.h. nicht mit `ReadFile()`).

1.19 **writeword**

Syntax

```
WriteWord(Number)
```

Beschreibung

Schreibt einen Word-Wert in die aktuelle Datei. Die Datei muß mit Schreib-Unterstützung geöffnet worden sein (d.h. nicht mit `ReadFile()`).