

**ADDENDUM aux**



**TRAVAUX PRATIQUES APPLESOFT**  
**La Programmation en BASIC avec ProDOS**



**ADDENDUM aux**

---

# **TRAVAUX PRATIQUES APPLESOFT**

---

**La Programmation en BASIC avec ProDOS**

---

---

---

# *Table des Matières*

<b>■ Chapitre 1</b>	<b>Introduction</b>	<b>5</b>
	5 Avant de commencer	
	6 Préparation d'une disquette d'amorçage ProDOS	
<b>■ Chapitre 2</b>	<b>La Syntaxe</b>	<b>7</b>
	7 Les noms de fichiers	
	7 Les noms d'accès	
	8 La syntaxe des commandes	
	8 Liste des options et des abréviations	
<b>■ Chapitre 3</b>	<b>Les Commandes ProDOS</b>	<b>11</b>
	11 Agencez vos fichiers	
	15 Manipulez vos fichiers	
	20 Dirigez et gérez vos entrées/sorties	
	21 Travaillez sur des fichiers texte	
	29 Manipulez des fichiers binaires	
<b>■</b>	<b>Bibliographie</b>	<b>31</b>

---

---

# Introduction

## ■ Avant de commencer

Ce supplément est destiné aux utilisateurs, qui après avoir fait leurs premiers pas en programmation, ont décidé d'aller un peu plus loin en utilisant les outils mis à leur disposition par ProDOS. Il est évident, qu'arrivé à ce niveau, les explications deviennent un peu plus techniques, mais si vous avez suivi la progression normale vous ne devriez pas rencontrer de problème.

ProDOS, si vous ne le savez pas, est le système d'exploitation de l'Apple II.

Qu'est ce qu'un système d'exploitation ?

C'est un programme en langage machine qui permet à une disquette de passer du stade de support magnétique, à celui de classeur pour toutes les informations que vous voulez stocker.

ProDOS met également à votre disposition toute une série de commandes utilisables en programmation BASIC qui vous permettent de commander ce classement et d'utiliser le mieux possible les informations stockées. ProDOS fait de votre disquette un meuble de rangement et vous laisse la liberté de l'agencement. La structure du meuble est le volume c'est-à-dire la disquette. ProDOS vous laisse mettre de 0 à 51 étagères, ce sont les catalogues. Ces étagères sont divisibles en sous-étagères, ce sont les sous-catalogues eux-mêmes divisibles, etc... Les niveaux de division peuvent être variables et à votre choix, à tout moment vous pouvez changer l'agencement initial, mais cela peut prendre un certain temps. Ce que vous mettez dans ces divisions d'étagères, ce sont vos fichiers sans aucune contrainte de taille si ce n'est celle du meuble. Certains fichiers sont actifs (programmes) d'autres sont passifs (fichiers texte, fichiers de données...).

Il ne reste plus maintenant qu'à exploiter cet espace disponible.

---

## **Préparation d'une disquette d'amorçage ProDOS**

Une disquette d'amorçage ProDOS est une disquette formatée qui contient le système d'exploitation.

Pour en préparer une il suffit donc, à l'aide de votre disquette utilitaire de formater une disquette et d'y recopier dessus les fichiers ProDOS et BASIC. SYSTEM qui sont déjà sur la disquette où se trouvent les utilitaires.

Vous détenez alors une disquette qui amorce le système dès la mise sous tension.

Si vous désirez qu'un programme particulier soit exécuté lors du démarrage du système il suffit de le sauver sous le nom de STARTUP.

# La Syntaxe

## Les noms de fichiers

Un nom de fichier sous ProDOS peut comporter jusqu'à 15 caractères (majuscules, minuscules, chiffres, points de ponctuation) et il doit obligatoirement commencer par une lettre, une minuscule étant automatiquement transformée en majuscule.

Un nom de fichier doit être unique dans un catalogue donné.

## Les noms d'accès

Un nom d'accès sous ProDOS est une série de noms de fichiers séparés par un slash (/). Il indique à ProDOS le chemin à parcourir pour parvenir au fichier désigné. Le premier nom de fichier dans un nom d'accès est le nom du volume où se trouve le fichier. Les noms suivants correspondent à la succession des divisions de division (catalogues, sous-catalogues, sous sous...) Pour terminer par le nom du fichier proprement dit.

La longueur maximum d'un nom d'accès utilisé dans une commande est de 64 caractères, slashes inclus.

Exemple :

```
✓PERSONNEL✓MARKETING✓PUBLICITE✓
```

Un nom d'accès partiel est un nom d'accès amputé de sa partie gauche et ne commençant pas par un slash.

Exemple :

```
MARKETING✓PUBLICITE✓ ou PUBLICITE✓
```

Toute commande vous demandant un nom de fichier acceptera un nom d'accès ou un nom d'accès partiel. Quel qu'il soit, sa longueur maximum devra être de 64 caractères slashes inclus.

Quand vous utilisez un nom d'accès partiel, ProDOS ajoute automatiquement le préfixe qu'il a en mémoire de façon à constituer un nom d'accès complet. Si le préfixe n'est pas en mémoire ProDOS utilisera le préfixe du volume qu'il adresse par défaut.

Dans l'exemple précédent le préfixe aurait dû être :

```
✓PERSONNEL✓ ou ✓PERSONNEL✓MARKETING✓
```

---

## La syntaxe des commandes

La structure d'une commande ProDOS est un nom d'instruction suivi d'une liste d'options, comme suit.

```
INSTRUCTION na (,Ss) (,Dd) (,Ff)
```

Les éléments entre parenthèses sont optionnels. Ils sont aussi appelés paramètres. Toutes les commandes n'admettent pas tous les paramètres, mais les paramètres admis pour une commande peuvent apparaître dans un ordre quelconque à moins qu'un ordre ne soit spécifiquement imposé. D'une manière générale les options sont utilisées pour affiner la commande ou pour changer les valeurs utilisées par défaut.

Les majuscules et les virgules sont les caractères que vous devez effectivement taper, les minuscules sont là pour remplacer votre sélection.

---

## Liste des options et des abréviations

Certaines commandes ProDOS utilisent des options. Les lignes qui suivent en font le répertoire en les accompagnant d'une définition. Ces options sont introduites à ce niveau afin de prédéfinir la syntaxe des commandes ProDOS mais vous aurez certainement à vous y référer lorsque vous attaquerez la définition des commandes. Il est même conseillé de s'y reporter le plus souvent possible pour comprendre les diverses alternatives d'une même commande.

- na nom d'accès ou nom d'accès partiel. Voir la partie sur les noms d'accès.
- ,Ss numéro de port ou de connecteur d'interface, il est fixé à 6 sur l'Apple IIc.
- ,Dd numéro du lecteur de disquette qui est 1 ou 2; le lecteur intégré de l'Apple IIc est le numéro 1.
- ,Ff numéro de position. Ff est utilisé avec les commandes READ, WRITE, POSITION et EXEC pour les fichiers texte à accès séquentiel et direct. Le numéro de position désigne une rubrique dont la position dans le fichier est à f rubriques en avant de la position du pointeur du fichier (une rubrique est délimitée par deux retours chariot (CHR\$(13)). La valeur par défaut de f est 0 ce qui ne modifie pas la position du pointeur de fichier.

- ,Rr    numéro d'enregistrement. Ce paramètre est utilisé avec les commandes READ, WRITE et POSITION. Le paramètre r désigne le r ième enregistrement d'un fichier texte à accès direct. La valeur par défaut est la dernière valeur de r spécifiée pour ce fichier.
- ,Ll    longueur en octet. Avec les commandes APPEND et OPEN lors de la création d'un fichier à accès direct, le paramètre l indique la longueur de l'enregistrement, par défaut l est à 1. Si le fichier existe déjà la valeur par défaut est celle qu'il avait à la création du fichier. Quand il est utilisé avec les commandes BRUN, BLOAD et BSAVE, l spécifie le nombre d'octets à transférer, du fichier à la mémoire interne de l'APPLE ou vice-versa.
- ,Ee    fin d'adresse en RAM. Cette option est une alternative de l'option Ll pour les commandes BRUN, BSAVE et BLOAD, (si elles sont utilisées simultanément seule la deuxième option dans la liste sera retenue) ; e indique la dernière adresse de la zone mémoire servant au transfert du fichier. La commande BSAVE nécessite l'utilisation de l'une ou l'autre de ces deux options, et si ni l'une ni l'autre n'est utilisée avec les commandes BRUN et BLOAD les octets seront transférés en mémoire vive jusqu'à la fin du fichier.
- án    numéro de ligne. Avec les commandes RUN et CHAIN, n indique le numéro de la première ligne de programme à exécuter. Si la ligne spécifiée par n n'existe pas, l'exécution commencera à la première ligne existante suivante. Par défaut n est la première ligne du programme.
- Ttype    type de fichier. type est une abréviation à trois caractères du type de fichier spécifié par la commande (cf page 10).
- nump    port numéro. nump est utilisé avec les commandes PR£ et IN£, il peut avoir n'importe quelle valeur de 0 à 7. nump est le numéro du port contenant l'extension vers ou en provenance de laquelle les données sont susceptibles d'être transférées.

- ,Bb numéro d'octet (byte). La valeur par défaut de b est 0, il indique une position dans le fichier qui est la position courante plus b octets. Lorsqu'il est utilisé avec les commandes READ et WRITE il est évalué après l'option (,Ff). La valeur maximum de b est la longueur du fichier, la valeur minimum est 1.  
Avec les commandes BRUN, BLOAD et BSAVE, il est toujours utilisé par rapport au début du fichier.
- ,Aa adresse en RAM. a indique l'adresse de début d'une zone mémoire de l'Apple. Lorsqu'il est utilisé avec les commandes BRUN, BLOAD et BSAVE cette zone est celle où est (ou sera) le fichier à transférer, avec PR£ et IN£ c'est la zone où se situe le programme de gestion des entrées/sorties; Pour BLOAD et BRUN la valeur par défaut de a est celle qui a été utilisée lors de la sauvegarde (BSAVE). Pour PR£ et IN£ c'est l'adresse Cnump00 où nump est le numéro du port activé.

Table : abréviations des types de fichier

abréviation	type de fichier
DIR	catalogue
TXT	texte
BAS	programme Applesoft
VAR	variables applesoft
BIN	binaire
REL	relogeable
\$Fn	défini par l'utilisateur
SYS	fichier système de ProDOS
SYS	programme système ProDOS

# Les commandes de ProDOS

Vous pouvez utiliser toutes les commandes ProDOS à l'intérieur d'un programme (mode exécution différée) et de même, toutes les commandes ProDOS sont directement accessibles à partir du clavier (mode exécution immédiate), exceptées OPEN, READ, WRITE, APPEND et POSITION. Une commande ProDOS utilisée en mode exécution différée doit être précédée du caractère CTRL-D (CHR\$ (4)) qui doit être le premier caractère de la ligne. Voici la manière la plus simple d'insérer une commande ProDOS dans un programme.

```
10 D$ = CHR$ (4)
20 PRINT D$; "COMMANDE ProDOS"
```

Cette commande ProDOS sera exécutée lors de la mise en route du programme contenant ces lignes. C'est le mode exécution différée.

```
COMMANDE ProDOS
```

Cette commande sera exécutée dès que vous aurez tapé . C'est le mode exécution immédiate.

Les commandes WRITE, READ et APPEND restent actives jusqu'à la commande ProDOS suivante. Si vous voulez désactiver une de ces commandes sans utiliser une autre commande ProDOS vous pouvez utiliser la commande ProDOS nulle :

```
50 PRINT CHR$ (4)
```

---

## Agencez vos fichiers

Cette partie fait le descriptif des commandes de ProDOS vous permettant d'agencer vos fichiers.

---

### CATALOG et CAT (na) (,Ss) (,Dd)

Exemples :

```
CAT/JEUX/SOURIS
```

```
CATALOG
```

```
CAT, $B
```

Liste des fichiers du catalogue souris dans le volume jeux

Catalogue étendu des fichiers contenus dans le préfixe ProDOS

Catalogue des fichiers contenus dans les volumes des lecteurs connectés au port 6

Agencez vos fichiers

CAT et CATALOG permettent d'afficher à l'écran la liste des fichiers contenus dans le catalogue spécifié optionnellement par na, Ss et Dd. Si les options ne sont pas utilisées, la liste du catalogue désigné par le préfixe est affichée à l'écran.

CAT affiche à l'écran une liste réduite (40 colonnes), alors que CATALOG fournit tous les renseignements (80 colonnes) sur les fichiers catalogués, qui sont, de gauche à droite :

- un astérisque si le fichier est verrouillé (voir la commande LOCK)
- le nom du fichier
- l'abréviation en 3 lettres du type de fichier
- le nombre de blocs de 512 octets occupés par le fichier
- la date de la dernière modification (mo/jo/an/h/min/sec) (seulement les trois premiers sont fournis par CAT)
- la date de création du fichier (mo/jo/an/h/min/sec)
- la longueur du fichier en octet
- l'adresse du chargement si c'est un fichier binaire ou la longueur d'enregistrement s'il s'agit d'un fichier texte à accès direct.

Quand vous demandez un catalogue, le nombre de blocs occupés et le nombre de blocs restants sont affichés.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO BUFFERS AVAILABLE
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	FILE TYPE MISMATCH
FILE BUSY	

---

#### **PREFIX (na) (,Ss) (,Dd)**

Exemples :

```
PREFIX/PERSONNEL/MARKETING.
```

```
PREFIX ,D1 ,S6
```

```
PREFIX/
```

le préfixe ProDOS devient  
/PERSONNEL/MARKETING

le nom du volume désigné  
par le port 6 et le drive 1  
est pris comme préfixe

vide le préfixe

affiche le préfixe courant à l'écran.

#### PREFIX

Cette commande est normalement utilisée pour fixer le préfixe, et si elle est utilisée sans option le préfixe courant est affiché à l'écran. Si la commande est utilisée sans option dans un programme, la prochaine instruction INPUT lira le préfixe. Si un slash est utilisé à la place du nom d'accès le préfixe est vidé et en cas de commande ProDOS le préfixe du volume pris par défaut sera utilisé. La longueur maximum d'un préfixe est de 64 caractères, slashes inclus.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	

---

#### **CREATE na (, Ttype), (,Ss) (,Dd)**

Exemples :

créé un nouveau catalogue, PARIS dans le volume ANNUAIRE

```
CREATE/ANNUAIRE/PARIS, TDIR
```

créé un fichier binaire dans le volume désigné par le port 6 et le lecteur 2.

```
CREATE IMAGE.0, D2, S6, TBIN
```

La commande CREATE permet de créer un fichier dans le volume, et du type, désignés par les options. Le fichier créé est vide mais il est désormais identifiable par ProDOS.

Vous devez d'abord créer un catalogue avant de sauver un fichier dedans.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	DIRECTORY FULL
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	DISK FULL
WRITE PROTECTED	DUPLICATE FILE NAME

---

### **RENAME na1, na2 (,Ss) (,Dd)**

Exemple :

```
RENAME EMPLOYES, PERSONNEL, S6, D1
```

La commande RENAME remplace le nom du fichier spécifié optionnellement par na1, Ss et Dd par na2. Cette commande ne peut pas être utilisée pour changer un fichier de catalogue. Elle ne fait que changer le nom sans toucher au fichier lui-même.

#### MESSAGES D'ERREUR POSSIBLES

```
RANGE ERROR          FILE BUSY
PATH NOT FOUND       FILE LOCKED
INVALID OPTION       NO DEVICE CONNECTED
SYNTAX ERROR         I/O ERROR
WRITE PROTECTED     DUPLICATE FILE NAME
```

---

### **DELETE na (,Ss) (,Dd)**

Exemple :

```
DELETE /PROGRAMMES/ESSAIS
```

Cette commande efface le fichier exprimé par na, Ss, Dd. Le fichier ne doit ni être verrouillé ni être ouvert (dans le cas d'un fichier texte). Pour effacer un catalogue, celui-ci doit être vide. Il n'est pas possible d'effacer un volume avec cette commande.

#### MESSAGES D'ERREUR POSSIBLES

```
RANGE ERROR          FILE BUSY
PATH NOT FOUND       FILE LOCKED
INVALID OPTION       NO DEVICE CONNECTED
SYNTAX ERROR         I/O ERROR
WRITE PROTECTED
```

### **LOCK na (,Ss) (,Dd)**

Exemple :

```
LOCK/PROGRAMMES/VERSION FINALE
```

Cette commande verrouille et protège le fichier spécifié optionnellement par na, Ss, Dd, d'un effacement ou d'un écrasement accidentel. Un fichier verrouillé présente un astérisque lors de l'affichage du catalogue à l'écran.

#### MESSAGES D'ERREUR POSSIBLES

```
RANGE ERROR          WRITE PROTECTED
PATH NOT FOUND       NO DEVICE CONNECTED
INVALID OPTION       I/O ERROR
SYNTAX ERROR
```

---

### **UNLOCK na (,Ss) (,Dd)**

Exemple :

```
UNLOCK/RECETTES/VIANDES
```

Cette commande permet de déverrouiller un fichier qui avait été préalablement verrouillé, afin de l'effacer, de le renommer ou de changer la version.

#### MESSAGES D'ERREUR POSSIBLES

```
RANGE ERROR          WRITE PROTECTED
PATH NOT FOUND       NO DEVICE CONNECTED
INVALID OPTION       I/O ERROR
SYNTAX ERROR
```

---

## **Manipulez vos fichiers**

Les commandes qui suivent vous permettent divers transferts, d'un volume à la mémoire vive et vice-versa.

---

charge le programme  
PERSONNEL du volume  
GESTION et l'exécute

---

### **- na (,Ss) (,Dd)**

Exemple :

```
-/GESTION/PERSONNEL
```

Cette commande appelée la commande "tiret" peut être utilisée à la place de RUN, BRUN et EXEC. C'est la seule commande permettant de charger puis d'exécuter un programme de type système (Ttype SYS), elle permet également d'exécuter un programme de type BAS, BIN ou TXT.

Notez que lors d'un appel de programme système toute la mémoire vive de l'Apple est employée, donc ce qu'elle contenait au moment de l'appel est perdu.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE
NO BUFFERS AVAILABLE	FILE(S) STILL OPEN

exécute le programme qui  
se trouve en mémoire vive

charge le programme TEST  
qui se trouve dans le  
volume MENU et l'exécute

charge le programme TEST  
qui se trouve dans le  
volume MENU et l'exécute  
à partir de la ligne 1000

---

### **RUN na (,án) (,Ss) (,Dd)**

Exemples :

```
RUN
```

```
RUN/MENU/TEST
```

```
RUN/MENU/TEST, d 1000
```

Cette commande charge le programme spécifié optionnellement par na, Ss et Dd, en mémoire vive et l'exécute à partir de la ligne désignée par n. La commande RUN sans aucune option exécute le programme BASIC qui se trouve en mémoire vive.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE

---

#### **LOAD na (,Ss) (,Dd)**

Exemple :

```
LOAD/ANNUAIRE/ACTUALISATION
```

charge dans la mémoire vive de l'Apple le programme BASIC ACTUALISATION qui se trouve dans le volume ANNUAIRE

Cette commande spécifie à ProDOS de rechercher le programme BASIC Applesoft (Ttype BAS) désigné par na, Ss et Dd, si ce fichier existe ProDOS le charge dans la mémoire vive de l'Apple, permettant ainsi de travailler dessus (correction, liste, exécution). Notez que le programme qui se trouvait au préalable dans la mémoire vive est alors effacé.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE

---

#### **SAVE na (,Ss) (,Dd)**

Exemple :

```
SAVE/DEMO/GRAPHIQUE
```

sauve le programme BASIC Applesoft qui se trouve en mémoire vive sur le volume DEMO sous le nom GRAPHIQUE

---

Cette commande transfère sur disque le programme BASIC qui se trouve en mémoire vive, dans le fichier désigné par na, Ss et Dd ; si ce fichier n'existe pas il est alors créé ; dans le cas contraire, le fichier existant est effacé et remplacé par celui qui se trouve en mémoire vive.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	FILE LOCKED
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	FILE TYPE MISMATCH
WRITE PROTECTED	DISK FULL
DIRECTORY FULL	

---

#### **CHAIN na (,án) (,Ss) (,Dd)**

Exemple :

```
CHAIN/EXEMPLES/PARTIE 2
```

Cette commande, qui lorsqu'elle est utilisée dans un programme, permet de charger en mémoire vive puis d'exécuter, le programme spécifié par na, Ss et Dd en préservant, les noms et les valeurs, des variables du programme utilisant cette commande. Ce qui veut dire qu'à l'aide de cette commande un programme peut opérer sur les résultats d'un précédent programme et laisser des données utilisables pour un nouveau chaînage.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE

stocke sur le volume EXPLOITATION sous le nom RESULTATS.VAR toutes les variables du programme BASIC Applesoft qui est en mémoire vive ainsi que leur valeur courante

---

## **STORE na (,Ss) (,Dd)**

Exemple :

```
STORE/EXPLOITATION/RESULTATS.VAR
```

Cette commande regroupe toutes les variables Applesoft définies par le programme qui se trouve en mémoire vive et stocke leur nom et leur valeur courante dans le fichier (Ttype VAR) spécifié optionnellement par na, Ss et Dd.

### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	FILE LOCKED
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	FILE TYPE MISMATCH
WRITE PROTECTED	DISK FULL
DIRECTORY FULL	

charge le fichier RESULTATS.VAR du volume EXPLOITATION dans la zone mémoire correspondant aux tableaux des variables définies par un programme Applesoft

---

## **RESTORE na (,Ss) (,Dd)**

Exemple :

```
RESTORE/EXPLOITATION RESULTATS.VAR
```

Cette commande efface les tableaux des variables Applesoft définies par le programme en mémoire vive, et charge à leur place le fichier désigné par na, Ss et Dd du type VAR et qui correspond à des variables Applesoft stockées préalablement par la commande STORE.

### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE

---

## **FRE**

Exemple :

```
10 PRINT CHR$(140); "FRE"
```

Lorsqu'un programme attribue une nouvelle chaîne de caractères à une variable chaîne préalablement définie, la nouvelle chaîne vient rallonger le tableau et l'ancienne persiste. Si un besoin de place en mémoire se fait sentir, il est alors nécessaire de faire le ménage dans ce tableau. La commande BASIC Applesoft FRE(X) assure ce ménage mais les routines de cette commande sont lentes alors que celles de la commande ProDOS FRE sont beaucoup plus rapides.

MESSAGES D'ERREUR POSSIBLES

```
SYNTAX ERROR
```

---

## **Dirigez et gérez vos entrées/sorties**

D'une manière simple, l'intérêt de l'Apple II et de tout ordinateur en général, est de prendre des données (ce sont les entrées) de travailler dessus, et de fournir des résultats (ce sont les sorties). Initialement, les entrées sont les données que vous tapez au clavier et les sorties sont les résultats que vous récupérez à l'écran. Avec l'Apple II il est possible de changer cette configuration de communication pour diriger par exemple les sorties vers une imprimante, un modem, etc... ou alors de spécifier à l'Apple qu'il ne doit plus prendre ses entrées à partir du clavier mais par exemple à partir d'un convertisseur, d'un modem, d'un lecteur de disquette, etc...

Les deux commandes qui suivent permettent de changer les directions des entrées/sorties.

les sorties seront dirigées vers le port 1

les sorties seront gérées par le programme binaire situé à l'adresse hexadécimale \$300

dirige les sorties vers le port 1 et désigne le programme situé à l'adresse hexadécimale \$C200 comme programme de gestion des sorties

---

### ***PR£ nump (,Aa) ou PR£, Aa***

Exemple :

```
PR£ 1
```

```
PR£ A$300
```

```
PR£ 1; A$C200
```

Cette commande est utilisée pour orienter les sorties vers le port désigné par nump (option nump seule), pour orienter les sorties vers une zone mémoire où se trouve un programme capable de les gérer (option Aa seule), pour orienter les sorties vers le port désigné par nump, en gérant le transfert par le programme situé à l'adresse a (option nump, Aa).

---

#### **▲ Attention**

Dans un programme PR£ doit être précédé de CTRL-D (CHR\$ (4)).

---

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR

SYNTAX ERROR

INVALID OPTION

NO DEVICE CONNECTED

---

### ***IN£ nump ou IN£ Aa***

Exemple :

```
IN£ 2
```

```
IN£ A$300
```

Cette commande prépare l'Apple II à recevoir ses entrées du port désigné par nump ou d'un programme situé à l'adresse a. Si les deux options sont utilisées simultanément l'Apple II reçoit ses entrées du port nump et les traite par le programme binaire situé à l'adresse a.

---

#### **▲ Attention**

Dans un programme IN£ doit être précédé de CTRL-D (CHR\$ (4)).

---

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR

SYNTAX ERROR

INVALID OPTION

NO DEVICE CONNECTED

Dirigez et gérez vos entrées/sorties

## Travaillez sur des fichiers texte

ouvre un fichier texte à accès séquentiel sous le nom de DATA dans la directory SESAME du volume désigné par D2 du port adressé par défaut

ouvre un fichier texte à accès direct sous le nom de OISEAU dans le volume CAGE et dont les enregistrements peuvent contenir 100 caractères

Ouvre le catalogue du nom MARKETING du volume PERSONNEL

### **OPEN na (,LI) (,Ttype) (,Ss) (,Dd)**

Exemples :

```
OPEN SESAME/DATA, D2
```

```
OPEN/CAGE/OISEAU, L100
```

```
OPEN/PERSONNEL/MARKETING, TDIR
```

Cette commande alloue une zone tampon en mémoire vive, au fichier spécifié optionnellement par na, Ss et Dd, de façon à travailler dessus. Si ce fichier n'existe pas, il est alors créé. L'option LI spécifie la longueur des enregistrement pour les fichiers à accès direct. Si l'option LI n'est pas utilisée et que le fichier n'existe pas, le fichier créé est un fichier à accès séquentiel.

Si l'option Ttype est utilisée le fichier désigné par na, Ss, Dd doit exister. Cette option vous permet alors de lire ou d'écrire dans un fichier qui n'est pas un fichier texte.

Vous devez être prudent lors de l'utilisation de cette option car le traitement des informations contenues dans un fichier qui n'est pas un fichier texte par des commandes BASIC peut être difficile.

Vous pouvez ouvrir jusqu'à 8 fichiers simultanément. Les commandes OPEN, CAT, CATALOG, EXEC et - (tiret) (quand elle est utilisée pour exécuter une commande EXEC), ouvrent toutes un fichier.

La zone mémoire tampon allouée à un fichier lors de son ouverture est de 1024 octets. Si il n'y a pas assez de mémoire libre pour qu'une zone tampon soit allouée, le fichier ne peut être ouvert.

### **▲ Attention**

Un programme doit fermer tous les fichiers qu'il a ouvert avant de rendre la main, dans le cas contraire les données écrites sur le fichier sont perdues.

## MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO BUFFERS AVAILABLE
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	FILE TYPE MISMATCH
WRITE PROTECTED	DUPLICATE FILE NAME
DIRECTORY FULL	NOT DIRECT COMMAND

---

### **CLOSE (na)**

Exemples :

```
CLOSE
```

```
CLOSE/LA.PORTE/SUP
```

La commande CLOSE sans option ferme tous les fichiers préalablement ouverts exceptés ceux ouverts par la commande EXEC. Si l'option na est utilisée, seul le fichier spécifié est fermé. Lorsque l'on ferme un fichier tous les caractères de la zone tampon allouée sont écrits dans ce fichier.

---

#### ▲ Attention

Si un programme s'arrête avant qu'il n'ait pu fermer tous les fichiers, il faut taper la commande CLOSE avant de commencer autre chose.

---

## MESSAGES D'ERREUR POSSIBLES

INVALID OPTION	I/O ERROR
SYNTAX ERROR	

---

### **READ na (,Rr) (,Ff) (Bb)**

La commande READ n'est utilisable qu'en mode exécution différée.

ferme tous les fichiers ouverts

ferme le fichier ouvert sous le nom de SVP dans le volume LA.PORTE

ouvre le fichier ANNUAIRE dans le volume MARSEILLE et se prépare à lire le dixième enregistrement

ouvre le fichier LIVRE, se déplace de 20 rubriques par rapport à la position courante et se prépare à lire

#### Exemples :

```
20 PRINT CHR$(4) ; "OPEN/ANNUAIRE/MARSEILLE"  
10 PRINT CHR$(4) ; "READ/ANNUAIRE/MARSEILLE,  
R10"
```

```
10 PRINT CHR$(4) ; "OPENLIVRE"  
20 PRINT CHR$(4) ; "READLIVRE, F20"
```

Cette commande prépare à la lecture de caractères en provenance du fichier désigné par na, à une position dans le fichier spécifiée optionnellement par Rr, Ff et Bb.

Une fois que cette commande est donnée tous les caractères demandés par les instructions INPUT et GET seront pris directement dans le fichier. La commande GET prend 1 caractère, INPUT prend tous les caractères jusqu'à ce qu'il trouve le code ASCII du retour chariot [(CHR\$(13))] ou que le nombre de caractères acquis atteigne 224.

Si vous ouvrez et lisez un catalogue (Ttype DIR), le texte lu aura le même format que les lignes fournies par CATALOG.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NOT DIRECT COMMAND
INVALID OPTION	FILE NOT OPEN
SYNTAX ERROR	END OF DATA
I/O ERROR	

---

#### **WRITE na (,Rr) (,Ff) (,Bb)**

La commande WRITE n'est utilisable qu'en mode exécution différée.

#### Exemples :

```
10 PRINT CHR$(4) ; "OPEN/ANNUAIRE/PARIS,R29"  
20 PRINT CHR$(4) ; "WRITE/ANNUAIRE/PARIS,R29"
```

ouvre le fichier PARIS dans le volume ANNAIRE et prépare à l'écriture au début de l'enregistrement 29.

ouvre le fichier DESSERTS  
dans le volume RECETTES  
se prépare à écrire à la  
position courante

```
10 PRINT CHR$(4) ; "OPEN/RECETTES/DESSERTS"  
20 PRINT CHR$(4) ; "WRITE/RECETTES/DESSERTS"
```

Cette commande prépare l'écriture à la position spécifiée  
optionnellement par Rr, Ff et Bb dans le fichier texte spécifié par  
na.

Une fois que cette commande est donnée, tous les caractères  
faisant l'objet d'une commande PRINT seront écrits directement  
dans le fichier.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	I/O ERROR
INVALID OPTION	NOT DIRECT COMMAND
SYNTAX ERROR	FILE NOT OPEN
WRITE PROTECTED	END OF DATA
FILE LOCKED	

---

### **APPEND na (,Ttype) (,LI) (,Ss) (,Dd)**

La commande APPEND n'est utilisable qu'en mode exécution  
différée.

Exemple :

```
10 PRINT CHR$(4) ; "APPEND/ASSOCIATION/ADRESSES"
```

Cette commande ouvre le fichier spécifié par na, Ss et Dd, et  
prépare à écrire à la fin du fichier. L'option LI est indispensable  
pour les fichiers à accès direct si on veut se positionner à la fin du  
fichier.

Une fois que cette commande est donnée, tous les caractères  
faisant l'objet d'une commande PRINT seront écrits dans la zone  
mémoire allouée au fichier ouvert.

---

#### ▲ Attention

Assurez-vous que votre programme referme bien tous les fichiers  
qu'il a ouverts avec la commande APPEND, sinon vous risqueriez  
de perdre des données.

---

ouvre le fichier ADRESSES  
dans le volume  
ASSOCIATION et se  
prépare à écrire à la fin du  
fichier.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
INVALID OPTION	PATH NOT FOUND
SYNTAX ERROR	FILE LOCKED
I/O ERROR	FILE TYPE MISMATCH
NO BUFFERS AVAILABLE	NOT DIRECT COMMAND
WRITE PROTECTED	

---

#### **FLUSH (na)**

Exemple :

```
FLUSH/POKER/DEAL
```

Cette commande, sans aucune option, provoque le transfert sur disque de toutes les zones tampon allouées aux fichiers préalablement ouverts. Si l'option na est utilisée, seule la zone tampon du fichier spécifié est transférée. Tout se passe comme pour la commande CLOSE à ceci près que les fichiers restent ouverts et que les pointeurs de fichiers ne sont pas altérés.

La commande flush est très utile afin de préserver l'intégrité des données sur disque. Un programme qui risque d'être interrompu accidentellement doit régulièrement utiliser la commande FLUSH de façon à perdre le moins de données possible en cas d'interruption.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	SYNTAX ERROR
INVALID OPTION	I/O ERROR

---

#### **POSITION na, Ff ou POSITION na ,Rr**

La commande POSITION n'est utilisable qu'en mode exécution différée.

Exemple :

```
10 PRINT CHR$(4); "OPEN COMMANDES"  
20 PRINT CHR$(4); "POSITION COMMANDES , F190"
```

Cette commande déplace le pointeur de fichier en fonction de la valeur des options Ff ou Rr.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NOT DIRECT COMMAND
INVALID OPTION	FILE NOT OPEN
SYNTAX ERROR	END OF DATA
I/O ERROR	

---

### **EXEC na (,Ff) (,Ss) (,Dd) ou EXEC na (,Rr) (,Ss) (,Dd)**

Exemple :

```
EXEC/RABLE/FONCTION, F3
```

exécute les commandes  
écrites dans le fichier  
FONCTION du volume  
RABLE à partir de la  
quatrième rubrique

Cette commande permet d'exécuter les commandes qui se trouvent dans le fichier spécifié par na, Ss et Dd en commençant à l'endroit spécifié par Ff ou Rr. Le fichier doit donc contenir à cet endroit une suite de commandes BASIC ou ProDOS écrites sous le format que vous utilisez lorsque vous les tapez au clavier. Le fichier spécifié doit être un fichier à accès séquentiel, les options Ff et Rr ont donc la même fonction qui est de positionner le pointeur de fichier au début de la rubrique désirée.

La procédure cataloguée permet de contrôler automatiquement le déroulement des opérations. La première rubrique est exécutée puis ainsi de suite jusqu'à la fin du fichier; à cet instant la commande EXEC referme le fichier. Il ne peut y avoir simultanément plusieurs commandes EXEC actives. Si le fichier EXEC contient une commande EXEC le premier fichier est alors fermé, le nouveau fichier EXEC est ouvert et exécuté. Une commande CLOSE provenant d'un fichier EXEC ne referme pas celui-ci. Si un fichier EXEC contient la commande RUN, ProDOS attend la fin d'exécution du RUN et passe à la rubrique suivante.

---

#### ▲ Attention

Si le fichier EXEC contient des commandes INPUT ou GET, il faudra prévoir que les caractères pris par ces commandes proviendront du fichier EXEC lui-même.

---

Si un programme est exécuté à partir d'une procédure EXEC il ne doit pas contenir les commandes INPUT ou GET, les caractères pris proviendraient de la rubrique suivante à exécuter et s'il s'agit d'une commande ProDOS à exécution immédiate, elle sera exécutée alors que le programme est en cours d'exécution. Les résultats peuvent être déroutants.

Une commande EXEC ne peut être interrompue par un CTRL-C.

---

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	NO BUFFER AVAILABLE

---

#### **Exemple d'écriture et de lecture d'un fichier texte**

Voici un petit programme qui vous montre la façon d'ouvrir un fichier texte à accès séquentiel et d'y écrire.

```
10 D$ = CHR$ (4)
20 PRINT D$; "OPEN LISTE ,S6, D1"
30 PRINT D$; "WRITE LISTE"
40 FOR I = 1 TO 10
50 PRINT "NOM"; I
60 NEXT I
70 PRINT D$; "CLOSE LISTE"
```

Et maintenant voici un programme permettant de lire le fichier créé précédemment.

```
10 D$ = CHR$ (4)
20 PRINT D$; "OPEN LISTE ,S6, D1"
30 PRINT D$; "READ LISTE"
40 FOR I = 1 TO 10
50 INPUT L$(I)
60 NEXT I
70 PRINT D$; "CLOSE LISTE"
80 FOR I = 1 TO 10
90 PRINT L$ (I)
100 NEXT I
110 PRINT "LISTE DES RUBRIQUES QUE CONTIENT
    LE FICHIER LISTE"
```

---

**BRUN na (,Aa) (,Bb) (,LI ou Ee) (,Ss) (,Dd)**

Exemple :

```
BRUN/DEMO/SOURIS ,A# 6000
```

Cette commande charge le fichier binaire spécifié par na, Ss et Dd (en prenant la partie du fichier spécifiée optionnellement par Bb et LI) dans la mémoire vive de l'Apple dans une zone définie optionnellement par Aa, Ee ou LI.

Une fois que le fichier (qui doit être un programme en langage machine) est chargé suivant les spécifications ci-dessus, la commande exécute le programme à partir de l'adresse Aa.

Si la dernière instruction est l'instruction RTS du 6502, l'Applesoft reprend la main pour exécuter de nouvelles commandes.

**MESSAGES D'ERREUR POSSIBLES**

```
RANGE ERROR           NO DEVICE CONNECTED
PATH NOT FOUND        I/O ERROR
INVALID OPTION       FILE TYPE MISMATCH
SYNTAX ERROR
```

---

**BSAVE na ,Aa, LI ou Ee (,Bb) (,Ttype) (,Ss) (,Dd)**

Exemples :

```
BSAVE IMAGE ,A#4000, E#5FFF
```

```
BSAVE IMAGE, A16304, L8192
```

La commande BSAVE stocke le contenu de la portion de mémoire délimitée par les options Aa, LI OU Ee et Bb dans le volume et sous le nom spécifié par na, Ss, Dd et de type Ttype. Par défaut le type de fichier est BIN. Si le fichier n'existe pas, il est créé. Aa spécifie l'adresse de début de la portion, LI sa longueur et Ee son adresse de fin. Si l'option Bb est utilisée le fichier sauvegardé commencera au b ième octet de la portion.

Les exemples ci-dessus provoquent tous les deux la sauvegarde de la portion de mémoire correspondant à la deuxième page graphique haute résolution. Ces commandes ont le même effet mais la première utilise une numérotation hexadécimale et l'option Ee, la deuxième utilise une numérotation décimale et l'option Ll.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	FILE LOCKED
PATH NOT FOUND	NO DEVICE CONNECTED
INVALID OPTION	I/O ERROR
SYNTAX ERROR	FILE TYPE MISMATCH
WRITE PROTECTED	DISK FULL
DIRECTORY FULL	

---

#### **BLOAD na (,Aa) (,Bb) (,Ll ou ,Ee) (,Ttype) (,Ss) (,Dd)**

Exemples :

```
BLOAD IMAGE ,A$192, L$192
```

```
BLOAD IMAGE, B$2000, A$4000, E$5FFF
```

La commande BLOAD charge le fichier spécifié par na, Ss et Dd (ou une partie si Bb est utilisée) dans une portion de mémoire vive (délimitée optionnellement par Aa, Ee ou Ll).

Dans les exemples ci-dessus le fichier IMAGE contient au moins 2 pages graphiques hautes résolutions toutes deux longues de 8192 octets. Le premier exemple place les 8192 premiers octets du fichier IMAGE dans la zone mémoire correspondant à la première page graphique haute résolution de l'Apple et le deuxième exemple place les 8192 octets suivants dans la portion de mémoire correspondant à la deuxième page graphique haute résolution.

#### MESSAGES D'ERREUR POSSIBLES

RANGE ERROR	NO DEVICE CONNECTED
PATH NOT FOUND	I/O ERROR
INVALID OPTION	FILE TYPE MISMATCH
SYNTAX ERROR	PROGRAM TOO LARGE

# ***Bibliographie***

Ce document est un supplément qui doit vous permettre d'effectuer vos premiers pas de programmation BASIC sous ProDOS. Il est certain que rapidement vous aurez besoin d'en savoir plus sur ProDOS. La librairie Apple met à votre disposition toute une série de manuels qui répondront à toutes les questions que vous serez amené à vous poser.

A2D2039 APPLESOFT PROGRAMMER'S KIT  
A2W0010 PRODOS TECHNICAL REFERENCE MANUAL  
A2L2008F MANUEL DE REFERENCE DU BASIC APPLESOFT  
A2L2005F MANUEL DE REFERENCE DE L'APPLE IIe  
A2W0013 PRODOS ASSEMBLER TOOLS  
A2D2010 PRODOS USER'S KIT



Avenue de l'Océanie  
Z.A. de Courtabœuf - B.P. 131  
91944 LES ULIS CEDEX  
Tél. : 33 (6) 928.01.39 - Télex 692719