

**TANDY®\***

**Système de  
disque  
d'ordinateur  
couleur**

**Guide de  
rapide  
référence**

\* MARQUE DE COMMERCE DE TANDY CORPORATION

*Guide de référence rapide de  
BASIC sur disque d'ordinateur couleur*  
Droit d'auteur 1987, InterTAN Canada Limitée,  
Barrie (Ontario)  
Tous droits réservés.

La reproduction ou l'utilisation, sans l'autorisation écrite expresse de Tandy Corporation de toute partie du présent manuel est interdite. Nous avons apporté le maximum de soins dans la préparation de ce manuel pour assurer sa précision, mais Tandy Corporation rejette toute responsabilité découlant de toute erreur ou omission ou de l'utilisation des renseignements qu'il contient.

# **BASIC sur disque d'ordinateur couleur**

**Guide de  
référence  
rapide**



# Introduction

---

Quand on relie une unité de disque souple à un ordinateur couleur, on commence automatiquement en BASIC sur disque à la mise en marche de l'ordinateur. Observer l'écran pour s'assurer que l'on a bien branché la cartouche d'interface de disque et l'unité de disque. Si tout est en ordre, l'écran affiche le message de droit d'auteur "DISK EXTENDED COLOR BASIC". Si ce n'est pas le cas, l'écran donne le message "EXTENDED COLOR BASIC".

Le présent guide résume les commandes disponibles avec le BASIC sur disque. Il donne aussi les messages d'erreur éventuels.



# Commandes de BASIC sur disque

## **BACKUP unité source TO unité de destination**

Reproduit le contenu du disque de l'unité source sur le disque de l'unité de destination. Si l'on n'a qu'une unité, elle doit être l'unité source.

```
BACKUP 0 TO 1   BACKUP 0
```

## **CLOSE # tampon,...**

Arrête la communication vers les tampons spécifiés. Si l'on omet le tampon, l'ordinateur ferme tous les fichiers ouverts.

```
CLOSE #1   CLOSE #1, #2
```

## **COPY "nom de fichier 1" TO "nom de fichier 2"**

Copie le contenu du nom de fichier 1 dans le nom de fichier 2. Chaque nom de fichier doit comprendre une extension.

```
COPY "FILE/BAS" TO "NEWFILE/BAS"  
COPY "ORG/DAT:0" TO "ORG/DAT:1"
```

## **CVN (variable en chaîne)**

Convertit une chaîne codée de 5 octets (créée à l'aide de MKN\$) au nombre qu'elle représente.

```
X=CVN (A$)
```

## **DIR numéro d'unité**

Affiche un répertoire du disque dans l'unité spécifiée.

```
DIR0   DIR
```

Exemple d'affichage:

```
MYPROG   BAS   0   B   3  
YOURPROG BAS   0   A   1  
HERDATA  DAT   1   A   5  
USPROG   BIN   2   B   2
```

De gauche à droite, les colonnes contiennent:

- Le nom de fichier
- L'extension
- Le type de fichier
  - 0 = programme BASIC
  - 1 = fichier de données BASIC
  - 2 = fichier en langage machine
  - 3 = fichier de source d'édition
- Le format de stockage
  - A = ASCII, B = binaire
- La longueur du fichier en granules

## **DOS**

Avec le minidisque de système OS-9 dans l'unité 0, la commande DOS amorce le système d'exploitation OS-9.

```
DOS
```

**DRIVE numéro d'unité**

Change l'unité par défaut à l'unité que l'on spécifie. Si l'on n'indique pas la commande DRIVE, l'ordinateur utilise l'unité 0 par défaut.

DRIVE 1

**DSKI\$ numéro d'unité, piste, secteur, variable en chaîne 1, variable en chaîne 2**

Entre les données d'un secteur donné, d'une piste donnée du disque, dans l'unité que l'on spécifie.

DSKI\$ 0, 12, 3, M\$, N\$

**DSKINI numéro d'unité**

Structure un disque dans l'unité que l'on spécifie. L'exécution de la commande efface la mémoire.

DISKINI 0 DISKINI 1

**DSKOS\$ numéro d'unité, piste, secteur, chaîne 1, chaîne 2**

Ecrit des données en chaîne dans le secteur, piste et numéro d'unité que l'on spécifie.

DSKOS 0, 2, 1, "FIRST DATA,"  
"SECOND DATA"

**EOF (tampon)**

Retourne une valeur 0 s'il y a d'autres données à lire dans le tampon et une valeur -1 s'il n'y en a plus.

IF EOF (1)=-1 THEN CLOSE #1

**FIELD# tampon, longueur de zone AS nom de zone,...**

Organise l'espace dans un tampon à accès direct en zones. Spécifier la longueur et le nom de chaque zone.

FIELD #1, 10 AS A\$, 12 AS B\$, 5 AS C\$

**FILES nombre des tampons, longueur**

Indique à l'ordinateur le nombre de tampons à réserver en mémoire et le nombre total d'octets (longueur) à garder pour ces tampons. Si l'on ne spécifie pas la longueur, l'ordinateur réserve un total de 256 octets.

FILES 1, 1000 FILE 5

**FREE (numéro d'unité)**

Retourne le nombre de granules libres du disque dans l'unité que l'on spécifie.

PRINT FREE (0)

**GET #tampon, numéro de fiche**

Tire la fiche suivante ou celle que l'on spécifie et la place dans le tampon.

GET #1, 5 GET #2, 3



**INPUT # tampon, nom de variable,...**

Entre les données du *tampon* que l'on spécifie et attribue chaque élément de donnée dans le *tampon* au *nom de variable* spécifié.

```
INPUT 1, A$, B$
```

**KILL "nom de fichier"**

Supprime le *nom de fichier* que l'on spécifie du répertoire de disque. On doit inclure l'extension avec le *nom de fichier*.

```
KILL "FILE/BAS", KILL "FILE/DAT:1"
```

**LINE INPUT # tampon, données**

Entre une ligne (toutes les *données* jusqu'au caractère ENTER) du *tampon* que l'on spécifie.

```
LINE INPUT #1, X$
```

**LOAD "nom de fichier", R**

Charge le fichier de programme BASIC spécifié en mémoire, à partir du disque. Si l'on inclut le R, l'ordinateur exécute le programme immédiatement après l'avoir chargé.

```
LOAD "PROGRAM", R  
LOAD "ACCTS/BAS:1"
```

**LOADM "nom de fichier", adresse de décalage**

Charge le fichier spécifié de programme en langage machine à partir du disque. On peut spécifier une *adresse de décalage* pour l'ajouter à l'adresse de chargement du programme.

```
LOADM "PROG/BIN", 3522
```

**LOC (tampon)**

Retourne le numéro de fiche courante du tampon que l'on spécifie.

```
PRINT LOC(1)
```

**LOF (tampon)**

Retourne la fiche de numéro le plus élevé du tampon que l'on spécifie.

```
FOR R=1 TO LOF(1)
```

**LSET nom de zone = données**

Justifie à gauche les données dans la zone que l'on spécifie.

```
LSET A$="BANANAS" LSET B$=T$
```

**MERGE "nom de fichier", R**

Charge le fichier de programme spécifié à partir du disque et le fusionne avec le programme en mémoire. Si l'on inclut le R, l'ordinateur passe le programme immédiatement après la fusion.

```
MERGE "SUB/BAS" MERGE "NEW", R
```

### **MKN\$ (nombre)**

Convertit le *nombre* spécifié en chaîne codée de 5 octets pour le stockage dans un fichier de disque structuré.

```
LSET B$=MKN$ (53678910)
```

### **OPEN "mode", #tampon, "nom de fichier", longueur de fiche**

Ouvre un tampon qui assure le transfert des données avec un dispositif particulier. Si l'on ne spécifie pas la *longueur de fiche*, l'ordinateur utilise 256 octets.

#### **Mode Fonction permise**

- I Entrée des données à partir d'un fichier à accès séquentiel.
- O Sortie des données vers un fichier à accès séquentiel.
- D Transfert des données vers un fichier à accès direct ou en provenance de ce fichier.

#### **Tampon Communication avec**

- 2 L'imprimante
- 1 Le magnétocassette
- 0 L'écran ou l'imprimante. (Il n'est pas utile d'ouvrir ce tampon.)
- 1-15 Les unités de disque.

```
OPEN "D", #1, "FILE", 15
```

```
OPEN "I", #2, "CHGE/DAT"
```

### **PRINT #tampon, liste de données**

Imprime les *données* dans le *tampon*. Séparer chaque élément de la liste à l'aide d'une virgule ou d'un point-virgule.

```
PRINT #1, "DATA"
```

**PRINT #tampon, USING format; liste de données**

Imprime les *données* dans le *tampon*, à l'aide du *format* spécifié. Le *format* est une chaîne; le mettre entre guillemets.

Les commandes de *format* sont:

#	Garde un espace pour un chiffre.
.	Imprime un point décimal.
,	Imprime une virgule immédiatement devant chaque groupe de trois chiffres (en comptant vers la gauche à partir du point décimal).
**	Remplit les espaces significatifs d'astérisques.
\$	Imprime un signe de dollar à gauche.
\$\$	Imprime un signe de dollar flottant.
+	Imprime le signe du nombre. Pour imprimer le signe devant le nombre, mettre le signe + au début de la chaîne de format. Pour imprimer le signe à la suite du nombre, mettre le signe + à la fin de la chaîne de format.
^^^	Imprime le nombre sous forme exponentielle.
-	Imprime un signe moins après le nombre si celui-ci est négatif. Cette commande n'imprime pas de signe si le nombre est positif. Mettre le signe moins à l'extrémité de droite de la chaîne de format.
!	Imprime le premier caractère de la chaîne.
%espaces%	Fixe la zone de la chaîne. La longueur de la zone correspond au nombre d'espaces, plus 2.

```
PRINT #1, USING "##.##";53.76
PRINT #2, USING "***$#.##-";-3.678
PRINT #1, USING "!"; "WHITE"
```

**PUT #tampon, numéro de fiche**

Attribue un *numéro de fiche* aux données du *tampon* que l'on spécifie. Si l'on ne spécifie pas de *numéro de fiche*, l'ordinateur l'attribue à la fiche courante.

```
PUT #2, 3 PUT #1, 4
```

**RENAME "ancien nom de fichier" TO "nouveau nom de fichier"**

Change le nom d'un fichier de disque. On doit spécifier l'extension des deux noms de fichier.

```
RENAME "MFILE/DAT:1"
TO "BFILE/DAT:1"
```

## **RSET *nom de fichier* = données**

Justifie à droite les *données* dans la zone que l'on spécifie.

```
RSET M$="SOAP"
```

## **RUN "*nom de fichier*", R**

Charge le *nom de fichier* à partir du disque et l'exécute. Si l'on inclut le R, tous les fichiers qui sont ouverts le restent.

```
RUN "FILE" RUN "PROG/BAS", R
```

## **SAVE "*nom de fichier*", A**

Sauvegarde le *nom de fichier* sur disque. En utilisant l'option A, on sauvegarde le programme sous format ASCII.

```
SAVE "PROG/BAS" SAVE "TEST:1", A
```

## **SAVEM "*nom de fichier*", première adresse, deuxième adresse, troisième adresse**

Sauvegarde le *nom de fichier* correspondant au programme en langage machine résidant dans la position de mémoire qui commence à la *première adresse* et se termine à la *deuxième*. La *troisième adresse* commande l'exécution.

```
SAVEM "FILE/BIN:1", &H5200,  
&H5800, &H5300
```

## **UNLOAD *numéro d'unité***

Ferme les fichiers ouverts du disque, dans l'unité que l'on spécifie. Si l'on ne spécifie pas de *numéro d'unité*, l'ordinateur utilise l'unité 0 (ou celle spécifiée dans la commande DRIVE).

```
UNLOAD 0 UNLOAD
```

## **VERIFY ON VERIFY OFF**

Met la fonction de vérification en marche ou l'arrête. Quand VERIFY est en marche, l'ordinateur vérifie toutes les écritures du disque.

```
VERIFY ON
```

## **WRITE #*tampon*, liste de données**

Écrit les *données* dans le *tampon* que l'on spécifie. Séparer les éléments de données de la liste par des virgules.

```
WRITE #1, A$, B$, C
```

## Messages d'erreur

---

<i>Erreur N°</i>	<i>Description</i>
/0	Division par zéro
AE 33	Fichier déjà existant
AO	Essai d'ouverture d'un fichier de données déjà ouvert
BR 27	Mauvais numéro de fiche
BS	Mauvais indice inférieur
CN	Impossible de continuer
DD	Essai de redimensionnement d'un tableau
DF 28	Disque plein
DN	Erreur de numéro d'unité ou de numéro de dispositif
DS	Instruction directe
ER 37	Ecriture ou entrée après la fin d'une fiche (accès direct seulement)
FC	Appel de fonction interdite
FD	Mauvaise donnée de fichier
FM	Mauvais mode de fichier
FN 31	Mauvais nom de fichier
FO 34	Dépassement de capacité de zone
FS 32	Mauvaise structure de fichier
HP	Erreur d'impression à haute définition (ordinateur couleur 3 seulement)
HR	Erreur de graphiques à haute définition (ordinateur couleur 3 seulement)
ID	Instruction directe interdite
IE	Entrée après la fin du fichier
IO	Erreur d'entrée/sortie
LS	Chaîne trop longue
NE 26	Fichier de disque introuvable
NF	NEXT sans FOR
NO	Fichier non ouvert
OB 29	Espace en dehors du tampon
OD	Fin de données
OM	Fin de mémoire
OS	Fin d'espace de chaîne
OV	Dépassement de capacité
RG	RETURN sans GOSUB
SE 35	Placement dans chaîne sans zone
SN	Erreur de syntaxe
ST	Formule de chaîne trop complexe
TM	Erreur d'assortiment de frappe
UL	Ligne non définie
VF 36	Vérification
WP 30	Disque protégé en écriture

FABRIQUÉ SPÉCIALEMENT POUR  
RADIO SHACK DIVISION, InterTAN CANADA LTÉE.  
BARRIE, ONTARIO. L4M 4W5

8A7  
Cat. No. 26-3133

Imprimé en Corée