

SMARKY NEWS

Le 15 mai 1987

No 42

Nouveautés dans le BASIC par Michael Walz

La révision 8.17 du BASIC comporte quelques nouveautés intéressantes, notamment des instructions qui permettent d'atteindre les bases de données créées avec FICHE.

1. Appels FICHE

Le format des appels est le suivant: FICHE (ORDRE,CANAL) où ORDRE est le numéro de l'ordre à effectuer et CANAL le canal fiche; 5 canaux peuvent être ouverts simultanément. Les fiches sont représentées dans le BASIC par des chaînes de caractères où chaque rubrique, y compris la dernière, est terminée par le caractère `␣` (CHR\$(13))

Exemple de fiche:

```
NM Dupont␣  
PR Daniel␣  
AD Ch. de la Cure 17␣  
VI Lausanne␣  
NP 1003␣
```

2. Ordres FICHE

Fiche (1,canal), FICHER\$, RESERVE

Ouvre la base de données dont le nom est dans la variable FICHER\$. Il n'est pas nécessaire de spécifier l'extension. RESERVE doit contenir le nombre de bytes que fiche réservera en plus. Si on omet RESERVE, c'est la librairie FICHE qui réserve elle-même un certain nombre de bytes.

Fiche (2,canal)

Ferme une base de données ouverte. La base de données n'est pas mise à jour.

Fiche (3,canal), CRITERE\$, INDEX%, MODE%, ELEMS

Cet appel peut effectuer deux choses différentes:

- La recherche d'une fiche d'après une fiche critère contenue dans CRITERE\$. Dans ce cas la recherche est faite à partir de la fiche dont le numéro est dans INDEX%. Attention: la recherche n'est effectuée que dans les fiches qui répondent à la sélection courante. MODE% n'est pas utilisé pour l'instant. La fonction `Fiche(2,canal)` peut alors retourner l'index de la fiche trouvée.

Exemple d'une fiche critère:

NM A*
VI Lausanne

- Si CRITERES\$ contient une chaîne vide, la fiche dont le numéro est dans INDEX% est lue avec la sélection courante (voir plus bas).

Fiche (4, canal), NO%

Utilise la sélection dont le numéro est dans NO% comme sélection courante. La première sélection porte le numéro 0. Si NO% contient -1 on accède aussi aux fiches systèmes.

Fiche (5, canal), ELEM\$, INIT\$, RUBR\$

Extrait de la fiche contenue dans ELEM\$ la rubrique dont les initiales sont dans INIT\$, et les met dans RUBR\$.

Exemple:

Soit dans ELEM\$ la fiche:

NM Dupont
PR Daniel
VI Lausanne

Si INIT\$ contient "VI", RUBR\$ contiendra "Lausanne"

Fiche (6, canal), ELEM\$, INIT\$, RUBR\$

Met à jour la rubrique de ELEM\$ dont les initiales sont dans INIT\$. Le nouveau contenu de la rubrique se trouve alors dans RUBR\$.

Exemple:

Soit dans ELEM\$ la fiche:

NM Dupont
PR Daniel
VI Lausanne

Fiche (7, canal), INDEX%

Détruit la fiche dont l'index est spécifié. Attention: l'index dépend de la sélection courante.

Fiche (8, canal), ELEM\$

Ecrit la fiche contenue dans ELEM\$ dans la base de données.

Fiche (9, canal), FICHER\$, SEL%

Met à jour la base de données dans le fichier dont le nom est contenu dans FICHER\$. Le fichier doit déjà exister, et avoir une longueur nulle. Si SEL% contient -1 toutes les fiches seront mises à jour, mais si SEL% contient un numéro de sélection, seules les fiches qui répondent au critère de sélection seront sauvées.

Fiche (10, canal), SEL\$, NO%

Permet d'ajouter une nouvelle sélection contenue dans SEL\$ et dont le numéro est NO%. La syntaxe de la sélection est décrite en détail dans la nouvelle brochure FICHE. On peut aussi créer des sélections avec le programme FICHE et consulter le fichier .FICHE avec EDIT pour visionner les sélections.

Fiche (11, canal), NUM%

Supprime la sélection dont le numéro est contenu dans NUM%.

3. Fonctions FICHE

Fiche (1, canal)

Retourne la dernière erreur survenue avec un appel FICHE. En effet une erreur comme: "Fiche pas trouvée" ne provoque pas d'erreur dans le BASIC. Seules les erreurs fatales telles que "Fichier n'existe pas", "Mémoire pleine", etc. sont détectées par le BASIC.

Fiche (2, canal)

Retourne l'index de la dernière fiche lue.

Fiche (3, canal)

Retourne la longueur de la dernière fiche lue.

Fiche (4, canal)

Retourne le nombre total de fiches contenues dans la base de données.

Fiche (5, canal)

Retourne le nombre de fiches selon la sélection actuelle.

Fiche (6, canal)

Retourne le nombre de bytes utilisés par les fiches.

Fiche (7, canal)

Retourne le nombre total de bytes à disposition des fiches.

La nouvelle brochure FICHE contient davantage de précisions.

Petit exemple

```
100 -- explore le fichier TEST.FICHE --
110 Fiche (1,1), 'TEST', 1000      -- ouvre le fichier TEST.FICHE
120 I=0                            -- index = 0
130 While Fiche(1,1)=0            -- tant qu'il n'y a pas d'erreur
140   Fiche (3,1), '', I, 0, E$    -- lit une fiche dans E$
150   I=I+1                        -- incrémente l'index
160   Print E$                     -- imprime la fiche
170 End While                      -- reboucle
180 Print Error$(Fiche(1,1))       -- imprime l'erreur FICHE
190 Fiche                          -- ferme la base de données
200 End
```

4. Souris et clavier

La fonction `Key` permet maintenant de lire simultanément le clavier *et* la souris. Pour cela l'ordre `Key` a été créé.

`Key (0)`

La fonction `Key` détecte uniquement le clavier. C'est le mode par défaut.

`Key (1)`

la fonction `Key` détecte le clavier *et* les boutons de la souris. Les codes pour les boutons sont les suivants:

```
32513 bouton de gauche pressé
32529 bouton de gauche relâché
32514 bouton du milieu pressé
```

32530 bouton du milieu relâché
32516 bouton de droite pressé
32532 bouton de droite relâché

Remarque: Les codes sont les mêmes que pour `Edit Line` en mode `Mouse Off`.

Key (2) Comme `Key(1)` mais les mouvements de la souris et les changements des touches supershift sont également détectés. Le code pour le mouvement de la souris est 32512.

Dans le cas de changement de touches supershift, la valeur rendue par `Key` est négative et le nouvel état des touches supershift est donné par l'expression: `HIGH(KEY)-128`.

Pkey Cette fonction rend le pseudocode de la dernière touche lue avec `Key`. Le pseudocode permet par exemple de distinguer si le `⌘` a été pressé sur le clavier ou sur le pavé numérique.

Voici un exemple:

```
100 Eop : Cursor Off : Mouse Off : Key(2)
110 Loop
120 A=Key      -- A contient le code de la touche pressée
130 Select A
140   Case 32513 : Print "Gauche pressé"
150   Case 32529 : Print "Gauche relâché"
160   Case 32512 : Print "Souris bougée"
170   Case 32514,32530,32516,32532 -- ne fait rien
180   Case Else
190     Do If A<0
200       Print "Changement supershift : ";High(A)-128
210     Do Else
220       Print "Touche: ";Low(A);" Supershift: ";High(A);" Pseudo: ";Pkey
230     End Do
240   End Select
250 End Loop
```

5. Exécution de programmes

L'ordre `Program` a été modifié afin qu'on puisse lancer un programme depuis le BASIC, sans que le BASIC attende la fin de ce programme.

Program A\$

Exécute le programme dont le nom est dans A\$, et attend sa fin.

Program A\$,0

Exécute le programme dont le nom est dans A\$ et assigne le clavier à ce programme, mais l'exécution du BASIC se poursuit.

Program A\$,1

Exécute le programme dont le nom est dans A\$, laisse le clavier assigné au BASIC et poursuit l'exécution du BASIC.

Un essai vous convaincra !