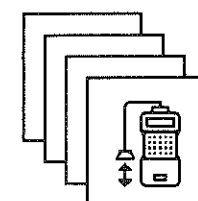


# Bulletin Technique

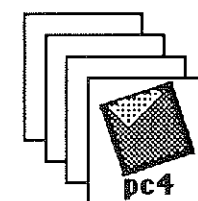
Janvier Février Mars 1989

La revue du support technique d'Aware



## ORGANISEUR II

- Etiquettes du fichier MAIN
- Personnaliser votre alarme
- Codes barres et programmes TV
- Animations graphiques
- Le signe ":" et l'OPL
- Supprimer l'option FIN
- Consommation automobile
- Compte à rebours



## PC 4

- Restructuration de fichiers
- Saisin et Saisist

## DERNIERES VERSIONS

Edité par : Aware  
7/9, rue des Petites Ecuries  
75010 PARIS  
Tél. : (1) 45.23.21.12  
Télex : 281 941  
Fax : (1) 45.23.02.37



★ A W A R E - P 8 9 - 1 ★

## Un colosse qui cache bien son jeu

A l'occasion du Sicob 89, nous présenterons un Organiseur dédié aux applications professionnelles : le POS 350.

Vu de loin, il ressemble à un XP : gris, 2 lignes d'écran, même disposition des touches du clavier, 2 slots.

Mais quand on appuie sur la touche ON, les choses se gâtent. Le POS 350 répond 'Pas de Pack', puis s'éteint.

En effet, cette machine ne dispose pas des fonctionnalités classiques de l'Organiseur II. Pas de Calepin, pas d'Agenda, pas d'Alarmes. Le POS 350 ne peut fonctionner que si on lui ajoute, dans l'un des slots, un Datapak contenant une procédure OPL appelée BOOT.

Si une telle procédure existe, le POS 350 l'exécute.

Il s'agit donc d'une machine dédiée à une ou plusieurs applications spécifiques.

Comparé au POS 200 et POS 250, cette machine offre les mêmes possibilités. A une différence près :

le POS 350 dispose de 96 Ko de RAM !

Sur ces 96 Ko de RAM, 32 sont effectivement disponibles en mémoire vive, les autres 64 Ko se comportent comme zone de stockage de données et de procédures.

C'est un peu comme si vous disposiez d'un XP 32 Ko avec deux Rampak de 32 Ko chacun. Tout en conservant les deux slots disponibles pour des Datapak, des Rampak ou des logiciels.

Le POS 350 permettra à l'Organiseur II de répondre aux exigences des applications nécessitant une grosse capacité en mémoire centrale.

Disponible immédiatement en toute petite quantité en raison de la pénurie de mémoire, le POS 350 coûtera aux environs de 2500 F HT.

Merci de votre attention.

Yves-Marie Lebrech

Directeur Général

## IMPRESSION D'ETIQUETTES AVEC L'ORGANISEUR

L'Organiseur offre la possibilité de réaliser des fichiers d'adresses, ce qui permet, si vous êtes muni de Comms-Link et d'une imprimante série, d'éditer des étiquettes.

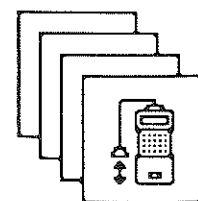
L'exemple suivant traite un fichier 'MAIN' qui ne contiendrait que trois lignes d'écritures au maximum par fiche.

EX:

AWARE  
7-9 rue des petites écuries  
75010 PARIS

Pour obtenir des étiquettes vous devez saisir la procédure qui suit en modifiant les valeurs pour l'adapter à vos besoins.

```
ETIQUET:
LOCAL A
CLS
OPEN "A:MAIN",A,A$,B$,C$
USE A
FIRST
DO
  A=0
  IF LEN(A.A$)<>0
    LPRINT A.A$
    A=A+1
  ENDIF
  IF LEN(A.B$)<>0
    LPRINT A.B$
    A=A+1
  ENDIF
  IF LEN(A.C$)<>0
    LPRINT A.C$
    A=A+1
  ENDIF
DEBUT::
  IF A<8
    A=A+1
    LPRINT
  ENDIF
  IF A<8
    GOTO DEBUT::
  ENDIF
NEXT
UNTIL EOF
CLOSE
```



Par  
Frédéric MARCHESI

2

### Modifications du nombre de lignes d'adresse :

4° Ligne :

(a\$,b\$,c\$) sont les noms des trois variables qui contiennent les lignes d'adresse. Si vous voulez rajouter des lignes vous devez ajouter autant de variables que de lignes d'adresse (EX:D\$ pour une ligne d'adresse supplémentaire).

9° à 12° ligne :

Teste la longueur de A.A\$; si sa valeur est différente de zéro l'imprimer sur l'étiquette, sinon passer à la 13° ligne.

A=A+1 (11° ligne) ajoute 1 au nombre de lignes imprimées (voir ligne 22).

13° à 16° ligne :

Test de la ligne d'adresse numéro 2.

17° à 20° ligne :

Test de la ligne d'adresse numéro 3.

Pour chaque ligne d'adresse que vous désirez ajouter vous devez ajouter un test

EX : Après la 20° ligne vous devrez insérer :

```
if len(A.D$)<>0
  lprint A.D$
  A=A+1
endif
```

pour une quatrième ligne d'adresse déclarée en D\$.

### Modifications de la taille des étiquettes :

21° à 28° ligne:

Teste sur le nombre de lignes déjà imprimées. Si le nombre de lignes est inférieur à 8 imprimer une ligne vierge jusqu'à ce que l'étiquette soit imprimée sur 8 lignes.

Si A=8 passer à l'enregistrement suivant (retour à la 7° ligne), remise à zéro de 'A' (ligne 8).

Vous devez régler le nombre de lignes imprimées par étiquettes en fonction de celles que vous possédez; n'oubliez pas de compter l'écart entre deux étiquettes comme une ligne.

Pour pouvoir imprimer vos étiquettes, n'oubliez pas de paramétrer la Comms Link en fonction de votre imprimante; reportez-vous au manuel d'utilisation de Comms Link.

PS : Si vous possédez une imprimante parallèle vous pouvez obtenir le même résultat en utilisant un convertisseur série/parallèle.

3

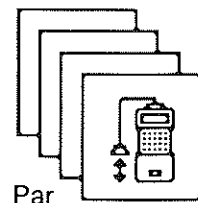
## PERSONNALISER VOTRE ALARME

On nous reproche souvent la faiblesse de l'alarme de l'organiseur II. Voici une procédure qui, quoiqu'un peu longue, permet de réaliser les sonorités que l'on désire obtenir (ah!...les goûts et les couleurs).

```

REVEIL:
local g%,f%,c%,p%,h%,m%,j%,mo%,a%,h$(2),m$(2),j$(2),a$(2),mo$(2)
rem g%=get
rem f%=frequence
rem c%=curseur
rem p%=position
rem j$ et j%=jour
rem mo$ et mo%=mois
rem a$ et a%=annee
rem h$ et h%=heure
rem m$ et m%=minute
j%=day
mo%=month
a%=year-1900
h%=hour
m%=minute+1
c%=1
p%=1
d::
rem*****Début de la boucle**
do
rem*****Contrôle de la validité**
if m%>59 :m%=0 :h%=h%+1 :elseif m%<0 :m%=59 :h%=h%-1 :endif
if h%>23 :h%=0 :j%=j%+1 :elseif h%<0 :h%=23 :j%=j%-1 :endif
if j%>31 :j%=1 :mo%=mo%+1 :elseif j%<1 :j%=31 :mo%=mo%-1 :endif
if mo%>12 :mo%=1 :a%=a%+1 :elseif mo%<1 :mo%=12 :a%=a%-1 :endif
if a%>99 :a%=99 :elseif a%<year-1900 :a%=year-1900 :endif
if m%<10 :m$="0"+gen$(m%,1) :else m$=gen$(m%,2) :endif
if h%<10 :h$="0"+gen$(h%,1) :else h$=gen$(h%,2) :endif
if j%<10 :j$="0"+gen$(j%,1) :else j$=gen$(j%,2) :endif
if mo%<10 :mo$="0"+gen$(mo%,1) :else mo$=gen$(mo%,2) :endif
a$=gen$(a%,2)
rem*****affichage**
cursor off
cls
print j$;"/";mo$;"/";a$,h$;":":m$
at c%,1
cursor on
g%=get
rem*****contrôle du clavier**
if g%=1
return

```



Par  
Gilles JEAN

```

elseif g%=5
if p%=1 :c%=13 :p%=5 :elseif p%=2 :c%=1 :p%=1
elseif p%=3 :c%=4 :p%=2 :elseif p%=4 :c%=7 :p%=3
elseif p%=5 :c%=10 :p%=4 :endif
elseif g%=6
if p%=1 :c%=4 :p%=2 :elseif p%=2 :c%=7 :p%=3
elseif p%=3 :c%=10 :p%=4 :elseif p%=4 :c%=13 :p%=5
elseif p%=5 :c%=1 :p%=1 :endif
elseif g%=3
if p%=1 :j%=j%+1 :elseif p%=2 :mo%=mo%+1
elseif p%=3 :a%=a%+1 :elseif p%=4 :h%=h%+1
elseif p%=5 :m%=m%+1 :endif
elseif g%=4
if p%=1 :j%=j%-1 :elseif p%=2 :mo%=mo%-1
elseif p%=3 :a%=a%-1 :elseif p%=4 :h%=h%-1
elseif p%=5 :m%=m%-1 :endif
elseif g%=13
else
beep 99,99
endif
until g%=13
cursor off
at 1,2
print"Valider 0/N"
g%=get
if g%=%n or g%=%N or g%=%$ or g%=1
goto d::
endif
rem*****programmation de l'alarme**
pokeb $22fe,0
pokeb $22f9,a%
pokeb $22fa,mo%-1
pokeb $22fb,j%-1
pokeb $22fc,h%
pokeb $22fd,m%-2
pokeb $22fe,1
f%=20
pokeb $00a4,1
off
pokeb $00a4,0
cls
print"Riser debout..."
while key<>13
rem*****musique**
do
beep 200,f%
f%=f%+5

```

```

until f%=150
do
beep 200,f%
f%=f%-5
until f%=20
endwh

```

Le principe de cette procédure est simple. Nous savons que la commande OFF permet d'éteindre l'Organiseur II, nous savons aussi qu'une alarme arrivée à terme rallume la machine et poursuit le programme en cours. Sachant qu'une alarme sonne pendant une minute, il suffit alors de programmer l'alarme aux adresses correspondantes en fonction de l'heure de réveil désirée moins une minute, de désactiver le buzzer et de l'éteindre (OFF). L'alarme ainsi programmée rallumera l'Organiseur et continuera votre programme musical préféré !...

## LES CODES BARRES ET LES PROGRAMMES TV

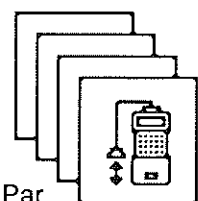
Je suis sûr que beaucoup d'entre vous ont pu remarquer les multiples campagnes de publicité pour les magnétoscopes pouvant être programmés à l'aide de codes barres. En effet, nombreux sont les amateurs de vidéo qui trouvent rébarbatif le système de programmation complexe permettant d'enregistrer des films. Deux solutions sont proposées :

- L'utilisation d'une carte sur laquelle on sélectionne les codes barres correspondant à la programmation choisie
- La lecture directe d'un code sur les magazines de programmes de télévision (ils sont de plus en plus nombreux à offrir ce service) ce qui est encore plus simple.

Dans les deux cas, un dispositif de télécommande à émission infra-rouges intègre un lecteur de codes barres qui peut lire le code sélectionné et transmettre au magnétoscope les informations nécessaires.

Nous allons nous intéresser plus particulièrement au second cas, de manière à vous expliquer comment cela fonctionne.

Prenons un exemple, le 24 Octobre dernier, Aware a organisé une grande fête dans ses nouveaux locaux, de 19 heures à 2 heures 30 du matin. La présentation de nos produits s'est déroulée entre 20 heures et 21 heures 30. Supposons que notre soirée intime (plus de 600 personnes étaient présentes) eût été retransmise en direct sur la 18 ème chaîne, le code barres que vous auriez trouvé dans votre magazine de programmes de télévision eut été le suivant :



Par  
Pierre LAGRANGE

Ce qui, et vous l'avez sans doute deviné, veut dire :

"Enregistrer le programme de la chaîne 18, le 24 octobre, à partir de 20 heures et jusqu'à 21 heures 30" (Véridique !)

En effet, le code barres que vous pouvez voir ci-dessus, donne, lorsqu'il est lu à partir d'un lecteur de codes barres, la série de chiffres suivante :

**18241020002130**

Et lorsqu'on découpe cette chaîne de chiffres en éléments de 2 chiffres, cela nous donne :

|    |                     |
|----|---------------------|
| 18 | Numéro de la chaîne |
| 24 | Jour du mois        |
| 10 | Mois                |
| 20 | Heure de début      |
| 00 | Minute de début     |
| 21 | Heure de fin        |
| 30 | Minute de fin       |

Vous voyez, ce n'est pas plus sorcier que cela. Mais, allez-vous me dire, pourquoi parler de ces codes barres dans le Bulletin Technique ?

Eh bien tout simplement parce qu'il est possible de lire ces codes barres à partir d'un Organiseur II, auquel on a ajouté un lecteur de codes barres sachant lire la plupart des codes utilisés actuellement. L'Organiseur ne peut pas transmettre les informations à votre magnétoscope, mais il ne faut pas désespérer, on pourrait imaginer un système de transmission à infra-rouges permettant de le faire. D'ailleurs, le projet est maintenant ouvert, et j'encourage toutes les personnes qui seraient tentées de développer un dispositif de ce type à se pencher sur la question.

Je vois même plus loin, car nous pouvons maintenant trouver sur le marché français des télécommandes dites universelles. Ces télécommandes sont en effet capable de transmettre des informations à la plupart des dispositifs à infra-rouges (téléviseurs, magnétoscopes, lecteurs de compact-disques, chaînes hi-fi, etc.). Mais ils sont encore plus intelligents, car ils sont capables d'apprendre les informations transmises par les télécommandes classiques, de les mémoriser, puis de les retransmettre à volonté, de manière à émuler la télécommande de base. Donc, si vous avez un super équipement, vous n'aurez plus à choisir parmi vos multiples télécommandes pour commander l'appareil voulu, mais à utiliser votre télécommande universelle, avec le canal approprié à l'appareil (Ces télécommandes universelles peuvent en général apprendre jusqu'à 8 appareils différents). Il serait donc sympathique d'imaginer un dispositif semblable avec l'Organiseur, c'est-à-dire un petit boîtier additionnel incorporant un système d'émission/réception infra-rouges et de lecture de codes barres. L'Organiseur II deviendrait alors le centre de pilotage de votre installation Hi-fi/vidéo.

C'est maintenant à vous de jouer, je suis ouvert à toutes les suggestions, mais contactez-moi uniquement par courrier, s'il vous plait. Je ne pourrai pas consacrer de temps au téléphone sur ce sujet.

## ANIMATION GRAPHIQUE SUR L'ORGANISEUR.

L'Organiseur permet de redéfinir huit caractères, comme l'expliquait l'article page 20 du Bulletin Technique N°35-36-37.

Sachant que l'Organiseur affiche à l'écran les caractères ASCII en fonction de la valeur des paramètres transmis à la procédure 'PARAM:' de l'article cité ci-dessus, en modifiant ces valeurs, on modifie l'affichage à l'écran sans repasser par un 'PRINT'.

Prenez comme exemple la procédure ci-dessous.

```
CROIX:
LOCAL A%
ESCAPE OFF
CLS
AT 1,1 :PRINT REPT$(CHR$(7),16)
AT 1,2 :PRINT REPT$(CHR$(7),16)
DO
A%=A%+1
IF A%=1 :PARAM:(7,0,0,0,31,0,0,0,0) :ENDIF
IF A%=2 :PARAM:(7,0,16,8,4,2,1,0,0) :ENDIF
IF A%=3 :PARAM:(7,0,4,4,4,4,4,0,0) :ENDIF
IF A%=4 :PARAM:(7,0,1,2,4,8,16,0,0) :ENDIF
IF A%=4 :A%=0 :ENDIF
UNTIL KEY<>0
```

8 Tapez ensuite la procédure PARAM:, si vous ne l'avez pas déjà dans la mémoire de votre Organiseur.

```
PARAM:(X%,A%,B%,C%,D%,E%,F%,G%,H%)
POKEB $180,64+X%*8
POKEB $181,A%
POKEB $181,B% :POKEB $181,C%
POKEB $181,D% :POKEB $181,E%
POKEB $181,F% :POKEB $181,G%
POKEB $181,H%
```

Une fois ces deux procédures saisies, vous pouvez lancer le programme en exécutant 'CROIX'.

Tapez une touche pour quitter le programme.

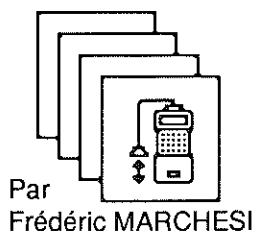
Cet exemple vous montre comment réaliser une animation graphique; vous pouvez l'utiliser pour réaliser quelques animations dans vos programmes.

## LE SIGNE ":" ET L'OPL

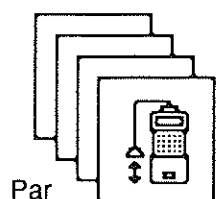
Voici un petit rappel qui mettra les choses au clair en ce qui concerne les différentes utilisations du signe ":" lors de la programmation en OPL.

En effet, il existe 3 possibilités d'utiliser ce caractère lorsque vous programmez en OPL, et celles-ci ont des significations très proches.

Nous allons illustrer cet article par une procédure OPL intégrant tous les cas de figure :



Par  
Frédéric MARCHESI



Par  
Pierre LAGRANGE

```
SIGNE2P:
LOCAL k%
DEBUT:: (a)
CLS :PRINT "IMPRESSION O/N" (b)
k%=GET
IF k%=%0
IMPRIME: (c)
ELSEIF k%=%N
STOP
ENDIF
GOTO DEBUT:: (a')
```

Cette procédure SIGNE2P: est un programme qui demande à l'utilisateur s'il veut effectuer une impression.

La réponse est saisie dans la variable k%. Sa valeur détermine la suite des opérations :

- L'utilisateur répond O pour OUI. Il est alors appelé la procédure IMPRIME: qui est exécutée.
- L'utilisateur répond N pour NON. Le programme est alors arrêté.
- L'utilisateur appuie sur une autre touche que O ou N. On se retrouve alors sur la ligne "GOTO DEBUT::", qui a pour effet de pointer la suite du programme sur l'étiquette DEBUT::. La question est alors à nouveau posée.

On peut remarquer dans cette procédure les 3 possibilités d'utilisation du signe ":".

(a) Signification d'une étiquette. Cela permet de marquer un endroit spécifique du programme, afin de pouvoir s'y brancher à un moment voulu (dans notre exemple par (a')).

Dans ce cas, il faut mettre deux fois le signe ":" à la suite du nom de l'étiquette, sans espace entre le nom et les signes.

(b) Séparation de plusieurs commandes indépendantes placées sur la même ligne. On peut placer autant de commandes indépendantes sur une même ligne, dans la limite de 254 caractères par ligne.

Dans ce cas, il faut obligatoirement placer un espace entre le dernier mot de la première commande et le signe ":". Il n'est pas obligatoire de placer un espace entre le signe ":" et le premier mot de la seconde commande.

(c) Appel d'une autre procédure.

Dans ce cas, il faut placer un signe ":" après le nom de la procédure que vous voulez appeler. Il est impératif de ne placer aucun espace entre le nom de la procédure et le signe ":".

Remarque : Si vous respectez bien les règles ci-dessus, vous ne rencontrerez plus de problèmes lors de l'élaboration de vos programmes.

## SUPPRIMER L'OPTION 'FIN' DE L'ORGANISEUR

L'Organiseur offre la possibilité de modifier le menu général pour lui donner l'aspect que l'on désire, mais pour des raisons de sécurité envers les personnes non initiées à la programmation, la seule option qui ne peut être retirée du menu principal est l'option 'FIN'.

On peut malgré tout retirer cette option par programmation, ce qui va permettre de réaliser une nouvelle commande d'extinction de l'organiseur, en y ajoutant toutes sortes d'ordres autant à l'extinction qu'à l'allumage.

Les trois procédures qui suivent permettent d'ôter une option du menu principal, si vous retirez la commande 'FIN', celle qui la précédait devient alors à son tour l'option protégée.

Tapez les trois procédures suivantes :

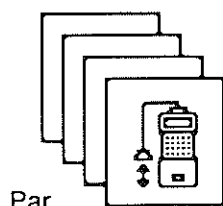
```
DELTOP:
LOCAL A$(8),F%
KSTAT 1
DO
F%=VIEW (1,"Nom de l'option à retirer")
IF F%<>8 AND F%<>13 AND LEN(A$)<=7
A$=A$+CHR$(F%)
ELSEIF F%=8 AND LEN(A$)>0
A$=LEFT$(A$,LEN(A$)-1)
ENDIF
AT 1,2:PRINT A$;" "
UNTIL F%=13
CALL:(ADDR(A$))

CALL:(1%)
LOCAL A%(2)
A%(1)=$183F
A%(2)=$6739
USR(ADDR(A%()),1%)

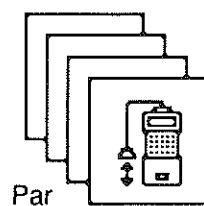
ENDFIN:(ITEM$,POS%)
LOCAL I%,A%(2)
POKEB $2187,LEN(ITEM$)
I%=1
WHILE I%<=LEN(ITEM$)
POKEB $2187+I%,ASC(MID$(ITEM$,I%,1))
I%=I%+1
ENDWH
POKEB $2188+LEN(ITEM$),0
A%(1)=$3F65
A%(2)=$3900
USR(ADDR(A%()),POS%)
```

Après avoir tapé ces trois procédures, il ne vous reste plus qu'à lancer le programme en tapant 'DELTOP', l'Organiseur va vous demander quelle option vous désirez retirer. En cas d'erreur il vous est possible de réinsérer l'option en suivant la procédure habituelle.

Vous pouvez maintenant vous réaliser vos propres programmes d'extinction.



Par  
Frédéric MARCHESI



Par  
Pierre LAGRANGE

## CONSOMMATION AUTOMOBILE

Voici un programme qui permettra à toutes les personnes qui utilisent un véhicule, d'en connaître la consommation moyenne aux 100 kilomètres.

Son principe en est simple. Lors de la première utilisation, le programme vous demande le kilométrage initial du véhicule ainsi que la date. Ces données doivent correspondre à un moment où le plein vient d'être fait. Ensuite, lors des utilisations ultérieures, vous aurez plusieurs choix proposés dans un menu : Saisie, pour entrer le nombre de litres pris au plein suivant, avec saisie du kilométrage correspondant. Il est alors affiché instantanément la consommation moyenne depuis le dernier plein.

Consulter, pour visualiser les différentes saisies et l'évolution des consommations. Il est possible d'aller à la première, précédente, suivante et dernière saisie.

Moyenne, pour connaître la moyenne générale de la consommation du véhicule pour l'ensemble des saisies.

Imprimer, pour imprimer sur l'imprimante Printer II de Psion la liste des saisies de consommation ainsi qu'un récapitulatif indiquant le kilométrage total et le nombre total de litres consommés, ainsi que la consommation moyenne générale.

Il est évidemment possible d'utiliser une autre imprimante que la Printer II, à condition de supprimer les codes de contrôle qui lui sont propres.

Quitter, pour quitter le programme.

```
CONSAUTO:
LOCAL m%,km,lt,kt,k%
TRAP CREATE "A:FCAUTO",A,d$,k,I :REM ***CREATION FICHIER
TRAP OPEN "A:FCAUTO",A,d$,k,I :REM ***OU OUVERTURE FICHIER
REM *****
REM ***DONNEES INITIALES DU FICHIER
IF EOF
a.d$=MID$(date$,1,15)
CLS
PRINT "Date initiale:"
EDIT a.d$
CLS
PRINT "Kilometrage ini-tial:";
INPUT a.k
a.l=0
APPEND
ENDIF
DO
REM *****
REM ***MENU GENERAL DU PROGRAMME
m%=MENU("Saisir,Consulter,Moyenne,Imprimer,Quitter")
IF m%=1 :REM ***SAISIE NOUVEAU PLEIN
LAST
km=a.k
```



```

CLS
a.d$=MID$(datim$,1,15)
PRINT "Date:"
EDIT a.d$
DO
CLS
PRINT "Kilometrage:"
INPUT a.k
REM ***TEST KILOMETRAGE CORRECT
IF a.k<=km
CLS
PRINT "Insuffisant"
PRINT "Dernier: ";km;"Km"
GET
ELSE
BREAK
ENDIF
UNTIL 0
CLS
PRINT "Litres:"
INPUT a.l
APPEND
REM ***CALCUL CONSOMMATION SUR DERNIERE SAISIE
CLS
km=a.k
BACK
km=km-a.k
NEXT
PRINT "Cons. moyenne:"
PRINT FIX$((a.l/km)*100,2,5),"L/100 Km"
GET
REM *****
ELSEIF m%=2 :REM *** CONSULTATION DES DONNEES
LAST
DO
KSTAT 1
IF POS=1
km=0
lt=0
ELSE
BACK
km=a.k
NEXT
km=a.k-km
lt=(a.l/km)*100
ENDIF
k%=DISP(1,CHR$(91)+"Db/Pr/Sv/Fn/Qt"+CHR$(93)+CHR$(9)+"Date:")

```

```

+a.d$+CHR$(9)+"Km Compt:"+GEN$(a.k,6)+chr$(9)+"Consomme:"
+GEN$(a.l,5)+" L"+CHR$(9)+"Parcours:"+GEN$(km,5)+" Km"
+CHR$(9)+"Moyenne:"+FIX$(lt,2,5)+" L/100")
IF k%=%D :REM ***PREMIERE FICHE
FIRST
ELSEIF k%=%P :REM ***FICHE PRECEDENTE
BACK
ELSEIF k%=%S :REM ***FICHE SUIVANTE
NEXT
IF EOF :REM ***RETOUR DERNIERE FICHE SI FIN DU FICHIER
LAST
ENDIF
ELSEIF k%=%F :REM ***DERNIERE FICHE
LAST
ENDIF
UNTIL k%=%Q :REM ***CONDITION SORTIE CONSULTATION
REM *****
ELSEIF m%=3 :REM ***CALCUL CONSOMMATION MOYENNE GENERALE
FIRST
lt=0
kt=a.k
do
lt=lt+a.l
NEXT
UNTIL EOF
LAST
kt=a.k-kt
CLS
PRINT "MOYENNE GENERALE"+FIX$((lt/kt)*100,2,5),"L/100 Km"
GET
REM *****
ELSEIF m%=4 :REM ***IMPRESSION DES DONNEES
REM LES LPRINT CHR$( ) SONT DES CODES DE CONTROLE DE LA PRINTER !!
LPRINT CHR$(23)+CHR$(21)+CHR$(17)
LPRINT "          Consommation automobile au",MID$(datim$,1,15),"
LPRINT
LPRINT " * DATE                      * KM CPT * LITR. * KM PAR * L/
100 *" +CHR$(24)
FIRST
DO
IF POS=1
km=0
kt=0
lt=0
ELSE
BACK
km=a.k-km
kt=kt+km

```

```

lt=lt+a,l
ENDIF
LPRINT "*",a,d$,"*",GEN$(a.k,-6),"*",
IF POS=1
LPRINT "      *      *      *"
ELSE
LPRINT GEN$(a.l,-5),"*",GEN$(km,-6),"*",FIX$((a.l/km)*100,2,-6),"*"
ENDIF
km=a.k
NEXT
UNTIL EOF
LPRINT "*-----*-----*-----*"
LPRINT CHR$(21)+"* Recapitulation *      *",GEN$(lt,-
5),"*",GEN$(kt,-6),"*",FIX$((lt/kt)*100,2,-6),"*"+CHR$(24)
ENDIF
UNTIL m%=5 :REM ***CONDITION SORTIE MENU GENERAL
REM *****
CLOSE :REM ***FERMETURE FICHIER

```

Remarque : pour fonctionner correctement, il faut toujours qu'une nouvelle saisie du kilométrage soit effectuée lorsque vous faites le plein du réservoir. En effet, le nombre de litres que vous saisirez correspondra au nombre de litres consommés depuis le dernier plein, pour la différence de kilomètres relevés entre ce plein et le précédent.

A vous de travailler, et bonne route...

## COMPTE À REBOURS

Creb est une petite procédure qui va vous permettre de décompter le temps de seconde en seconde.

Selon la durée du beep vous devrez modifier la longueur de la pause pour conserver un décomptage en secondes correctes, sans oublier de prendre en compte le nombre de tests effectués dans votre boucle.

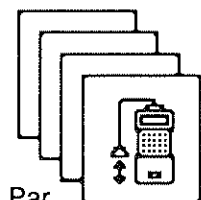
```

CREB:
LOCAL C,X
ESCAPE OFF
PRINT"TEMPS : "; INPUT C
PRINT"(Secondes)"
DO
PAUSE 18 :BEEP 100,500 :CLS
IF C>1 :X=14 :ELSE X=13 :ENDIF
PRINT LEFT$("*      Secondes",X) :AT 15,1 :PRINT" *"
AT 3,1 :PRINT C :C=C-1 :UNTIL C=-1
BEEP 300,300 :BEEP 200,800 :BEEP 300,300

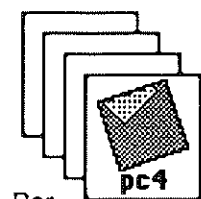
```

## RESTRUCTURATION DE FICHIER

PC Archive est une base de données permettant une manipulation aisée des informations saisies. Il met pour cela à votre disposition un panel de commandes très complet. Ce langage de programmation procédural dispose d'une



Par  
Olivier LAVIGNE



Par  
Gilles JEAN

syntaxe identique à la plupart des standards de développement existants. Il pourrait vous arriver de découvrir, au fur et à mesure que votre travail avance, qu'une rubrique manque cruellement à votre fichier. Voici une technique, autre que celle décrite dans le manuel (p144), permettant de mettre en avant toute la puissance du langage TSL (Task Sequencing Language) et permettant de rajouter une rubrique en fin de fichier. Le principe consistant, avant toute chose, à exporter votre fichier Archive par la commande 'exporte' et à quitter votre programme. Tapez par exemple le programme suivant sous Quill, faites un export du document en le nommant par exemple "convert.tsl" et quittez votre traitement de texte. De retour au DOS, tapez QUILL/TSL=CONVERT puis return.

```

[f3]
[f3]
d
&d Frappez le nom du document à importer suivi de RETURN
i
&i
&d
&d
&d
&d
[cr]
[f3]
p
f
[lt]
edo
,""
[rt]
[up]
[lt]
&d Transformation (T ou t pour valider ou ESCAPE pour sortir)
&g
&d
&d
&d
&d
&= t do
&= T do
&= [esc] fin
efin
&p 0
&e

```



Le TSL vous demande tout d'abord le nom du fichier à importer dans Quill (c'est le nom du fichier exporté avec Archive) puis se positionne automatiquement sur la fin de chaque ligne en commençant par le bas et vous demande de tapez la lettre t ou T pour insérer une nouvelle rubrique vide a l'enregistrement Archive (Car vous avez sans doute déjà remarqué que chaque ligne correspondait à un enregistrement différent de votre fichier mise à part la première qui représente la structure du fichier). Ceci étant, n'oubliez pas de saisir la nouvelle rubrique à la fin de la première ligne. Cet article n'a pour but que de vous montrer les immenses possibilités du langage TSL et de sa simplicité pour les mettre en oeuvre afin d'automatiser les taches répétitives.

UTILISATION DE 'SAISISN' ET 'SAISIST' DANS ABACUS.

Abacus permet par les commandes 'SAISISN' et 'SAISIST' de produire une ou plusieurs cellules du tableau qui seront définies à l'exécution des formules par les commandes '(F3) et E(pour Exec)'.

Comme exemple, j'ai créé un petit tableau qui :

|         |                                               |
|---------|-----------------------------------------------|
| Demande | Débit / Crédit.                               |
| Demande | Combien.                                      |
| Calcule | La somme en + ou - selon le mode d'opération. |
| Affiche | Le résultat en B3.                            |

pour créer ce tableau il faut insérer dans les cases :

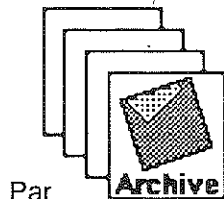
| Position | Valeur à saisir                        | saisie    |
|----------|----------------------------------------|-----------|
| A1       | Dernière opération                     | en texte  |
| A2       | Saisist("Débit / Crédit ")             | en valeur |
| A3       | saisien("Combien ")                    | en valeur |
| B1       | Solde en cours                         | en texte  |
| B2       | Solde                                  | en texte  |
| B3       | $B3+a3*(code(A2)=67)-A3*(code(A2)=68)$ | en valeur |

Une fois que vous aurez créé le tableau, lancez-le en tapant 'F3' et E(pour Executer).

A la première Question ( Débit / Crédit ), vous devez au moins saisir la première lettre en MAJUSCULE, car la cellule 'B3' teste la valeur ASCII de la première lettre pour définir si c'est un Débit ou un Crédit.

Dans la table ASCII :

|   |      |                 |
|---|------|-----------------|
| C | = 67 | alors que c=99  |
| D | = 68 | alors que d=100 |



Par  
Frédéric MARCHESI

Dernieres  
Versions

ORGANISEUR

|            |     |
|------------|-----|
| CM         | 3.3 |
| XP         | 3.6 |
| Filepak    | 1.4 |
| Barpak     | 0.9 |
| Tableur    | 1.2 |
| Diary Link | 1.0 |
| Formulat   | 1.0 |
| Maths      | 1.0 |
| Fin pak    | 1.0 |
| Top Fin    | 1.0 |
| Développe  | 2.2 |
| CL PC      | 2.0 |
| CL MAC     | 2.0 |

PC4

|     |     |
|-----|-----|
| Pc4 | 1.1 |
|-----|-----|