

TITLE Editeur, utilitaires <util.as>

*
 * Ce module contient les utilitaires propres à l'éditeur
 * et ses fonctions.
 * Les utilitaires pris dans JPC Rom sont dans jpcutil.as
 *

 * KEYMT
 *
 * But: attendre une pression de touche de l'utilisateur. Les
 * touches de curseur horizontal sont autorisées pour le
 * déroulement de l'affichage.
 * Entree: -
 * Sortie: -
 * - A(B) := B(B) := code physique (IOS2, p. 12-13)
 * - Cy = 1
 * Abime: A-D, D0, D1, P, D2 nbs at SCRTCH (CKSREQ)
 * Appelle: SCRLLR, CKSREQ, POPBUF, ATNCLR
 * Niveaux: 6
 * Detail: Cette routine n'est pas l'équivalent exact
 * de keyut dans KA car celle de KA détecte KOFF et
 * branche directement à l'extinction de KA.
 * Historique:
 * 86/08/01: ajout de documentation
 * 88/05/23: PD/JT pompage dans KA pour T/XEDIT

 * KEYWT COSBVL = SCRLLR Voir KEYWITS (JPC No 20), sauf que
 * 0030 0000 3F0000 GONC kut10 li, eh bien les touches de
 * 0031 0000 508 CONIC COSBVL = CKSREQ déroulement de l'affichage sont
 * 0032 0000 8F0000 COTO = KEYWT supportées.
 * 0033 0000 11EEF kut10
 * 0034 0000 3F0000 COSBVL = POPBUF
 * 0035 0000 8F0000 COSBVL = ATNCLR Abime A(A)
 * 0036 0000 D4 A+B A Pour compatibilité avec FINDA
 * 0037 0000 01 RTN

 * posmsg, poscmd
 * But: réaliser POS(MSG)(C), A)
 * Entree: -
 * - A(B) = caractère à chercher
 * - posmsg seulement :
 * C(A) = numéro de message
 * Sortie: -
 * - Cy = 0 : non trouvé
 * - Cy = 1 : trouvé
 * C(A) = D(A) = 0..LEN(MSGS(..)-1)*

0101 00062 962 7A=C B
 0102 00065 51 GOYES posm50 trouvé 1
 0103 00067 161 D0=D0+ 2
 0104 0006A E7 D=0+1 A
 0105 0006C 136 posm20 CDREX
 0106 0006F 134 D0=C
 0107 00072 801 7C=B A Peut-on continuer ?
 0108 00075 AE GOYES posm10 oui.
 0109 00077 400 RTNC pas trouvé (Cy = 0)
 0110 0007A
 0111 0007A DB posm50 C=D A C(A) := index
 0112 0007C 01 RTN trouvé (Cy = 1)
 0113 0007E
 0114 0007E
 0115 0007E
 0116 0007E
 0117 0007E
 0118 0007E
 0119 0007E
 0120 0007E
 0121 0007E
 0122 0007E
 0123 0007E
 0124 0007E
 0125 0007E
 0126 0007E
 0127 0007E
 0128 0007E
 0129 0007E
 0130 0007E 1800000 *PURGE D0=(5) *EXTFIL Destruction "interne" ?
 0131 00085 142 A=DAT0 A
 0132 00088 8A8 7A=0 A Pas de fichier externe
 0133 00088 00 RTNYES alors retour
 0134 0008D 850 7S=0 *SCREAT
 0135 00090 00 RTNYES Le fichier existait déjà
 0136 00092
 0137 00092 06 RSTK=C RSTK := C(A)
 0138 00094
 0139 00094
 0140 00094
 0141 00094 3470FFF
 0142 00098 EA A=A-C A A(A) := end of file
 0143 0009D 05 B=C A B(A) := offset (dest - source)
 0144 0009F 06 C=A A C(A) := end of file
 0145 000A1
 0146 000A1
 0147 000A1
 0148 000A1
 0149 000A1
 0150 000A1 8F00000 GOSBVL = MVHEM+
 *
 * [l ne peut pas y avoir d'erreur (Cy = 1) car
 * - le fichier est supprimé (donc suffisamment de mémoire)
 * - le fichier est créé (donc pas en Rom)
 *
 * C=RSTK
 * RTN

 * setfil
 *
 * But: mettre le flag 1 (le vrai flag 1, celui de
 * l'utilisateur) à l'état indiqué par C(0).
 * Entree: -
 * - C(0) = 0 ou #0
 * Sortie: -
 * - P = 0
 * Abime: A(S), C(S), D0, P
 * Appelle: 0
 * Niveaux: 0
 * Historique:
 * 88/11/12: PD/JT conception & codage

 * setfil D0=(5) *FLGREG D0 := " flags utilisateur
 * A=DAT0 S A(S) := flags 0 à 3
 * P= 15
 * LC(1) *FLMSK C(S) := FLMSK
 * P= 0
 * 7C=0 P setfil20 Mettre le flag à 0 ?
 * GOYES setfil20 oui
 * A=AIC S non : mettre le flag à 1
 * GOTO setfil30
 * setfil20 C=C-1 S C(S) := not FLMSK
 * A=AIC S
 * setfil30 DAT0=A S flags 0 à 3 := nouveaux flags
 * RTN

 * extsk, extsk+
 *
 * But: chercher des lignes dans un fichier externe.
 * Entree: -
 * - A(A) = adresse du fichier externe
 * - C(A) = numéro de ligne cible
 * - extsk+ seulement :
 * - B(A) = numéro d'une ligne (= C dans le fichier
 * - D0 = adresse de cette ligne

* Abime: A-D, D0, D1, R0, R1, R2 si message insert. OUTBS
 * Appelle: CONVUC, FPOLL, D0=AVS, TBMSG\$
 * Note: la routine "poscmd" est réservée à l'éditeur de
 * textes. Le numéro du message à chercher est codé en dur.
 * Note: la documentation de TBMSG\$ est archi-fausse:
 * - il faudrait lire :
 * * Exit:
 * ** D0 = " start of string
 * ** C(A) = B(A) = " end of string (final FF)
 *
 * Merci HP !
 * Niveaux: 3
 * Historique:
 * 88/10/29: PD/JT conception & codage

 * poscmd GOSBVL = CONVUC A(B) := commande en majuscule
 * 0066 00027 2F00000 LC(4) (=ld)*(*taVCHD)
 * 0067 0002E 330000 *posmsg R0=C R0(3-0) := numéro du message
 * 0068 00034 108 R1=A R1(B) := caractère à chercher
 * 0069 00037 101
 * 0070 0003A
 * 0071 0003A
 * 0072 0003A
 * Y-a-t-il un traducteur pour ce message ?
 * 0073 0003A 2F00000 COSBVL = FPOLL abime A(A)-D(A), D0, D1
 * 0074 00041 00 CON(2) = pTRANS abime A-D, D0, D1, P
 * 0075 00043
 * R0(3-0) = nouveau numéro de message s'il y a eu traduction
 * 0076 00043
 * R1(B) = caractère à chercher
 * 0077 00043
 * 0078 00043
 * 0079 00043 2F00000 COSBVL = D0=AVS D0 := (AVHEMS)
 * 0080 0004A 118 C=R0 C(A) := numéro de message
 * 0081 0004D 2F00000 GOSBVL = TBMSG\$
 * 0082 00054
 * 0083 00054
 * D0 = " début de la chaîne
 * C(A) = " fin
 * 0084 00054
 * R1(B) = caractère à chercher
 * 0085 00054
 * Modifie pour tenir compte des erreurs de la documentation
 * 0086 00054
 * de TBMSG\$.
 * 0087 00054
 * 0088 00054
 * 0089 00054 111 A=R1 A(B) := caractère à chercher
 * 0090 00057 05 B=C A B(A) := " fin du message
 * 0091 00059 03 D=0 A
 * 0092 0005B 6010 GOTO posm20
 * 0093 0005F
 * Invariant de boucle :
 * 0094 0005F
 * A(B) = caractère à chercher
 * 0095 0005F
 * D0 = " caractère du message à tester
 * 0096 0005F
 * D(A) = indice du caractère à tester (0..LEN-1)
 * 0097 0005F
 * B(A) = " terminateur de la table
 * 0098 0005F
 * 0099 0005F
 * 0100 0005F 14E posm10 C=DAT0 B

0151 000A8
 0152 000A8
 0153 000A8
 0154 000A8
 0155 000A8
 0156 000A8 07 C=RSTK
 0157 000AA 01 RTN
 0158 000AC
 0159 000AC
 0160 000AC
 0161 000AC
 0162 000AC
 0163 000AC
 0164 000AC
 0165 000AC
 0166 000AC
 0167 000AC
 0168 000AC
 0169 000AC
 0170 000AC
 0171 000AC
 0172 000AC
 0173 000AC
 0174 000AC
 0175 000AC 1800000 *setfil D0=(5) *FLGREG D0 := " flags utilisateur
 0176 000B3 1524 A=DAT0 S A(S) := flags 0 à 3
 0177 000B7 2F P= 15
 0178 000B9 300 LC(1) *FLMSK C(S) := FLMSK
 0179 000BC 20 P= 0
 0180 000BE
 0181 000BE 90A 7C=0 P setfil20 Mettre le flag à 0 ?
 0182 000C1 A0 GOYES setfil20 oui
 0183 000C3 9E4E A=AIC S non : mettre le flag à 1
 0184 000C7 6A00 GOTO setfil30
 0185 000CB BCE setfil20 C=C-1 S C(S) := not FLMSK
 0186 000CE 9E4E A=AIC S
 0187 000D2
 0188 000D2 1504 setfil30 DAT0=A S flags 0 à 3 := nouveaux flags
 0189 000D6 01 RTN
 0190 000D8
 0191 000D8
 0192 000D8
 0193 000D8
 0194 000D8
 0195 000D8
 0196 000D8
 0197 000D8
 0198 000D8
 0199 000D8
 0200 000D8

```

0201 00008      * Sortie:
0202 00008      * - A(A) = numéro de ligne trouvée (= C en entrée)
0203 00008      * - D0 = " ligne trouvée (longueur LIF)
0204 00008      * Abime: A-D, D0
0205 00008      * Appelle: -
0206 00008      * Niveaux: 1 (utilisé pour sauvegarde dans RSTK)
0207 00008      * Historique:
0208 00008      * 88/11/13: PD/JT extraction de D et généralisation
0209 00008      * 88/11/27: PD/JT modification de l'interface
0210 00008      *
0211 00008      *
0212 00008 130  *extsk D0=A      D0 := " header du fichier externe
0213 00008 16F  D0=D0+ 16  D0 := " file type
0214 00008 16F  D0=D0+ 16  D0 := " REL(5) FiLeNd
0215 00008 164  D0=D0+ 5   D0 := " début des données
0216 00008 01   B=0   A      B(A) := 1 (pointe première ligne)
0217 00008 E5   B=B+1  A      B(A) := 1 (pointe première ligne)
0218 00008      *
0219 00008      * A(A) = " file header
0220 00008      * B(A) = numéro d'une ligne référence
0221 00008      * D0 = " début de cette ligne
0222 00008      * C(A) = numéro de la ligne à trouver
0223 00008      *
0224 00008 D7   *extsk D=C   A      D(A) := numéro de ligne à chercher
0225 00008 136  CD0EX  C(A) := " ligne référence
0226 00008 06   RSTK=C  RSTK := " ligne référence
0227 00008      *
0228 00008      * A(A) = " file header
0229 00008      * B(A) = numéro de la ligne référence
0230 00008      * D(A) = numéro de la ligne à chercher
0231 00008      * RSTK = " ligne référence
0232 00008      *
0233 00008      *
0234 00008 D4   D0=A      D0 := " header
0235 00008 16F  A=B   A      A(A) := numéro de la ligne référence
0236 00008 16F  D0=D0+ 16  D0 := " REL(5) FiLeNd
0237 00008 136  CD0EX  C(A) := " REL(5) FiLeNd
0238 00008 134  D0=C      D0 := "
0239 00008 05   B=C   A      B(A) := " REL(5)
0240 00008 146  C=DATE  A      C(A) := REL(5)
0241 00008 C1   B=B+C  A      B(A) := " FiLeNd
0242 00008      *
0243 00008      * A(A) = numéro de la ligne référence
0244 00008      * B(A) = " FiLeNd
0245 00008      * D(A) = numéro de la ligne à chercher
0246 00008      * RSTK = " ligne référence
0247 00008      *
0248 00008 07   C=RSTK
0249 00008 134  D0=C      D0 := " ligne référence
0250 00008

```

```

0301 00148 C2      C=C+A  A
0302 00148 134  D0=C      D0 := " ligne suivante
0303 00150      *
0304 00150      * D0 = " ligne suivante
0305 00150      * RSTK = numéro de la ligne précédente
0306 00150      * B(A) = " FiLeNd
0307 00150      * D(A) = numéro de ligne à chercher
0308 00150      *
0309 00150      *
0310 00150 07   C=RSTK
0311 00152 E6   C=C+1  A      i++
0312 00154      *
0313 00154 88F  ske200 ?C=D  A      i >= ligne à chercher ?
0314 00157 70   GOYES  ske310 oui : on arrête
0315 00159 588  GONC   ske100
0316 0015C      *
0317 0015C      *
0318 0015C      * Sortie sur EOF
0319 0015C      * RSTK = i
0320 0015C      * D0 = " EOF
0321 0015C      *
0322 0015C 07   ske300 C=RSTK  C(A) := i
0323 0015E DA   ske310 A=C   A      A(A) := i
0324 00160 01   RTN
0325 00162      *
0326 00162      *
0327 00162      * YNQ
0328 00162      *
0329 00162      * But: prendre un code physique de touche, et tester si la
0330 00162      * touche est une de celles autorisées par le message
0331 00162      * *teKEYS, ce qui permet de traduire les réponses
0332 00162      * autorisées à une demande "Yes/No/Quit".
0333 00162      * Entrée:
0334 00162      * - A(B) = code physique de la touche (renvoyé par KEYWT)
0335 00162      * Sortie:
0336 00162      * - C(0) = 0 : non reconnu
0337 00162      * 1 : [Y]
0338 00162      * 2 : [N]
0339 00162      * 3 : [Q]
0340 00162      * Abime: A-C, R0-R1, S0-S2, D0, D1, OUTBS
0341 00162      * Appelle: KEYNAM, posmsg
0342 00162      * Niveaux: 4 (posmsg)
0343 00162      * Historique:
0344 00162      * 88/11/13: PD/JT conception & codage
0345 00162      *
0346 00162      *
0347 00162 8F00000 YNQ  COSBVL  *KEYNAM Conversion en nom de touche
0348 00169      *
0349 00169      * A(HP) = ASCII for keycode
0350 00169      * P = word thru pointer length of text

```

```

0251 0010C D6      C=A   A      C(A) := numéro ligne référence
0252 0010E      *
0253 0010E      * pour i := (ligne référence) jusque (ligne à chercher)
0254 0010E      * faire
0255 0010E      * si D0 >= FiLeNd ou D0 = JFFF
0256 0010E      * alors sortir de la boucle "pour"
0257 0010E      * fin si
0258 0010E      * passer à la ligne suivante
0259 0010E      * fin pour
0260 0010E      * renvoyer i
0261 0010E      *
0262 0010E      *
0263 0010E      *
0264 0010E      *
0265 0010E      * Affectation des registres pendant la boucle :
0266 0010E      * - B(A) = " FiLeNd
0267 0010E      * - C(A) = numéro de la ligne courante (i)
0268 0010E      * - D(A) = numéro de la ligne à chercher
0269 0010E      * - D0 = " ligne courante (longueur LIF)
0270 0010E      *
0271 0010E 6540  GOTO   ske200 aller directement au test sur i
0272 00112      *
0273 00112 06   ske100 RSTK=C  RSTK := i
0274 00114 136  CD0EX  C(A) := " ligne courante
0275 00117 134  D0=C      D0 := " ligne courante
0276 0011A 88D  ?C=B   A      et EOF ?
0277 0011D F3   GOYES  ske300 oui : D0 = " fin du fichier
0278 0011F      *
0279 0011F 15A3  A=DATE 4
0280 00123      * SUPBYT (#17A24) recopié pour éviter un COSBVL
0281 00123 D6   C=A   A
0282 00125 F0   ASL   A
0283 00127 F0   ASL   A
0284 00129 F6   CSR   A
0285 0012B F6   CSR   A      C(4) := 0 (entre autres)
0286 0012D AEA  A=C   B      A(A) := longueur LIF
0287 00130      * fin du pompage
0288 00130 23   P=   3
0289 00132 A96  C=A   UP  on sait que C(4) = 0
0290 00135 DA   A=C   A
0291 00137 814  A=A+1  UP  Cy = i si C(A) = FFFF
0292 0013A 20   P=   0
0293 0013C 4F1  GOC   ske300 D0 = " FFFF
0294 0013F      * conversion en nombre de quartets à sauter
0295 0013F 81C  ASRB
0296 00142 E4   A=A+1  A
0297 00144 C4   A=A+1  A
0298 00146 C4   A=A+1  A      A(A) := nb de quartets à sauter
0299 00148      *
0300 00148 136  CD0EX

```

```

0351 00169      *
0352 00169 881  ?P#   1      # 1 caractère ?
0353 0016C 72   GOYES  YN0500 non reconnu
0354 0016E 20   P=   0
0355 00170 8F00000 GOSBVL =CONVUC Conversion en majuscules
0356 00177 3300000 LC(4) (=id)(*teKEYS)
0357 0017D 73BE  GOSUB  *posmsg
0358 00181 511  GONC   YN0500 non trouvé
0359 00184 D0   A=0   A
0360 00186 E4   A=A+1  A
0361 00188 E4   A=A+1  A      A(A) := 2
0362 0018A 882  ?C=A  A
0363 0018D 60   GOYES  YN0500 Non reconnu ! Message erroné !
0364 0018F E6   C=C+1  A      renvoyer numéro dans (1..3)
0365 00191 01   RTN
0366 00193      *
0367 00193 D2   YN0500 C=0   A      C(0) := 0
0368 00195 20   P=   0
0369 00197 01   RTN
0370 00199      *
0371 00199      *
0372 00199      * CHKXTX
0373 00199      *
0374 00199      * But: vérifier que le fichier pointé par D1 est du type
0375 00199      * TEXT.
0376 00199      * Entrée:
0377 00199      * - D1 = " file header
0378 00199      * Sortie:
0379 00199      * - Cy = 0 : pas d'erreur
0380 00199      * D1 inchangé
0381 00199      * - Cy = 1 : ce n'est pas un TEXT
0382 00199      * C(3-0) = aFYPE
0383 00199      * Abime: A(A), A(S), C
0384 00199      * Appelle: -
0385 00199      * Niveaux: 0
0386 00199      * Historique:
0387 00199      * 88/11/27: PD/JT conception & codage
0388 00199      * 89/06/11: PD/JT reconception & recodage
0389 00199      *
0390 00199      *
0391 00199 17F  *CHKXTX D1=D1+ 16  D1 := " file type
0392 0019C D0   A=0   A
0393 0019E 1583  A=DATE 4      A(A) := file type
0394 001A2 D2   C=0   A
0395 001A4 E6   C=C+1  A      C(A) := fTEXT
0396 001A6 8A2  ?A=C  A
0397 001A9 A0   GOYES  CKXTX1 Ok
0398 001AB      *
0399 001AB      * Invalid File Type
0400 001AB      *

```

```

0401 001A8 330000      LC(4)  *FTYPE
0402 001B1 02          RTNSC      Cy = 1
0403 001B3
0404 001B3 LCF        CKTX1L D1=D1- 16   remet D1 à l'original
0405 001B6 03          RTNCC      Cy = 0
0406 001B8
0407 001B8
0408 001B8
0409 001B8
0410 001B8
0411 001B8
0412 001B8
0413 001B8
0414 001B8
0415 001B8
0416 001B8
0417 001B8
0418 001B8
0419 001B8
0420 001B8
0421 001B8
0422 001B8
0423 001B8
0424 001B8
0425 001B8
0426 001B8
0427 001B8
0428 001B8
0429 001B8
0430 001B8
0431 001B8 1F00000    D1=(5) *S-R0-0
0432 001B8 07          C+RSTK
0433 001C1 145        DAT1=C  A      S-R0-0 := sauvegarde d'un niveau
0434 001C4
0435 001C4
0436 001C4
0437 001C4 8F00000    GOSBVL *FSPECx 6 niveaux pour lui tout seul !
0438 001C8
0439 001C8
0440 001C8
0441 001C8 1F00000    D1=(5) *S-R0-0
0442 001D2 147        C+DAT1  A      C(A) := niveau sauvé au dessus
0443 001D5 06          RSTK=C
0444 001D7
0445 001D7
0446 001D7
0447 001D7 442
0448 001DA 978        GOC      bserr  Cy = 1 (<=> erreur dans l'analyse
0449 001D0 91          DAT1=A  W      A = 0 (<=> pas de nom de fichier
0450 001DF

```

```

0501 00228 A4E        C=C-1  S
0502 0022E 943        7C=D   S
0503 00231 96         GOYES  FL2-20 no device specified
0504 00233
0505 00233
0506 00233
0507 00233 871      7ST=1  1      fichier doit être copié en mémoire ?
0508 00236 A0         GOYES  FL2-10  oui
0509 00238
0510 00238 02        C=0    A
0511 00238 3100      LC(2)  *FSPEC      Cy = 1 : erreur
0512 0023E 92        RTNSC
0513 00240
0514 00240
0515 00240
0516 00240
0517 00240 101      FL2-10 R1=A
0518 00243 3123      LC(2)  50
0519 00247 8F00000  GOSBVL *SALLOc
0520 0024E
0521 0024E 111      * Abime : A, B(A), C(A), D1
0522 00251 843      A=R1
0523 00254 8F00000  ST=0   3      Source
0524 0025B
0525 0025B AF0      GOSBVL *SVINF+
0526 0025E D3        * Abime : A, C, D, R0, D1, S3, S4
0527 00260 853      A=0    W
0528 00263 8F00000  D=0    A
0529 0026A          ST=1   3
0530 0026A          GOSBVL *SVINFO
0531 0026A
0532 0026A
0533 0026A
0534 0026A
0535 0026A 07        C+RSTK      C(A) := adresse de l'appelant
0536 0026C DA        A=C      A
0537 0026E 8F00000  GOSBVL *PSHUPD
0538 00275
0539 00275 8F00000  GOSBVL *MGOSUB
0540 0027C 00000    CON(5)  *COPYU copie
0541 00281
0542 00281 441      * Abime : A-D, D0, D1, R0-R2, S0-S9, S12, STMTR0
0543 00284          GOC      Bserr  Erreur lors de la copie
0544 00284
0545 00284 8F00000  GOSBVL *POPUPD
0546 00288 DB        C=D      A      C(A) := adresse dépliée
0547 0028F          RSTK=C
0548 0028F 119      C=R1     C(A) =  début du fichier
0549 00292 6680      GOTO    FL2-40
0550 00295

```

```

0451 001DF
0452 001DF
0453 001DF 1F00000    D1=(5) *STMTR0
0454 001E6 AF8        C=D     W      Sauvegarde de D(W) pour CRETF+
0455 001E9 1557      DAT1=C  W
0456 001ED 17F       D1=D1- 16     D1=(5) STMTR1
0457 001F0 1517      DAT1=A  W      Sauvegarde du nom du fichier
0458 001F4 01        RTN
0459 001F6
0460 001F6 02        fspecx C=0   A
0461 001F8 3100      LC(2)  *FSPEC
0462 001FC 2D00000   bserr  GOVLNG  *BSERR
0463 00203
0464 00203
0465 00203
0466 00203
0467 00203
0468 00203
0469 00203
0470 00203
0471 00203
0472 00203
0473 00203
0474 00203
0475 00203
0476 00203
0477 00203
0478 00203
0479 00203
0480 00203
0481 00203
0482 00203
0483 00203
0484 00203
0485 00203
0486 00203
0487 00203
0488 00203
0489 00203 1F00000    D1=(5) *STMTR0
0490 0020A 1577      C+DAT1  W
0491 0020E AF7        D=C     W      Restaurer pour CRETF+
0492 00211 17F       D1=D1- 16     D1=(5) *STMTR1
0493 00214 1537      A+DAT1  W      Comme en sortie de FSPECx
0494 00218 AC2        C=0     S
0495 00218 943        7C=D    S
0496 0021E C7        GOYES  FL2-20 :HAIN
0497 00229 846        C=C-1  S
0498 00223 943        7C=D    S
0499 00226 47        GOYES  FL2-20 :PORT
0500 00228 AC2        C=0     S

```

```

0551 00296 656F      bserr  GOTO  bserr  Rallonge
0552 0029A
0553 0029A
0554 0029A
0555 0029A
0556 0029A 1F00000  FL2-20 D1=(5) *STMTR0
0557 002A1 AF8        C=D     W
0558 002A4 1557      DAT1=C  W
0559 002A8 17F       D1=D1- 16
0560 002AB 1517      DAT1=A  W
0561 002AF 8F00000  GOSBVL *FINDF
0562 002B6
0563 002B6 137      * Abime : A-D, S6, S8, R1-R3
0564 002B9 5F5      CDIEX
0565 002BC          GONC   FL2-40 C(A) := début du fichier
0566 002BC
0567 002BC
0568 002BC 872
0569 002BF 70        7ST=1  2      Faut-il le créer ?
0570 002C1          GOYES  FL2-30 Oui !
0571 002C1
0572 002C1
0573 002C1
0574 002C1 137      CDIEX      C(A) := numéro d'erreur
0575 002C4 02        RTNSC      Cy = 1 => il y a eu erreur
0576 002C6
0577 002C6
0578 002C6
0579 002C6 1F00000  FL2-30 D1=(5) *STMTR0
0580 002CD 1577      C+DAT1  W
0581 002D1 AF7        D=C     W
0582 002D4 D2        C=0     A
0583 002D6 3132      LC(2)  37+4
0584 002DA 8F00000  GOSBVL *CRETF+ Théoriquement, le Lex ne doit pas bouger
0585 002E1
0586 002E1 448      * Abime : A-D, D0, D1, R0-R1, S0-S7
0587 002E4 119      GOC      Bserr
0588 002E7 1F00000  C=R1
0589 002EE 1537      D1=(5) *STMTR1
0590 002F2 135      A+DAT1  W
0591 002F5 1517      D1=C    W
0592 002F9 17F       DAT1=A  W      Nom du fichier
0593 002FC 35100000  D1=D1- 16     D1 := " file type
0594 00304 15D5      LCHEX  400001 text + copy code
0595 00308 17F       DAT1=C  6
0596 00308 174      D1=D1- 16
0597 0030E D2        D1=D1- 5
0598 00310 CE        C=0     A      EOF mark
0599 00312 15D3      C=C-1  A
0600 00316 119      DAT1=C  4
0601 00316 119      C=R1     C(A) =  début du fichier

```

```

0601 00319      *
0602 00319      * Le fichier est trouvé. On a son adresse dans C(A)
0603 00319      *
0604 00319 03  FL2-48 RTNCC      Cy = 0 : pas d'erreur
0605 00318      *
0606 00318      *
0607 00318      *
0608 00318      *
0609 00318      *
0610 00318      *
0611 00318      *
0612 00318      *
0613 00318      *
0614 00318      *
0615 00318      *
0616 00318      *
0617 00318      *
0618 00318      *
0619 00318      *
0620 00318      *
0621 00318      *
0622 00318      *
0623 00318      *
0624 00318      *
0625 00318      *
0626 00318      *
0627 00318      *
0628 00318      *
0629 00318 8F00000  GOSBVL =ISRAM? Abime A, B(A), C, D1
0630 00322 09      C=B      A      C(A) := header sauve par ISRAM?
0631 00324 512     GONC      CKWrom
0632 00327      *
0633 00327      *
0634 00327      *
0635 00327 135     D1=C      D1 := " File header
0636 00328 8F00000  GOSBVL =GETPRI
0637 00331 09      C=B      A      C(A) := B(A) non modifié par GETPRI
0638 00333      *
0639 00333 832     SB = 1 si le fichier est SECURE
0640 00336 50      75B=0   Fichier non sécurisé
0641 00338 580     GOYES   CKW010
0642 00338      *
0643 00338      *
0644 00338      *
0645 00338      *
0646 00338 7730   GOSUB   CHKOPN
0647 00338 580     GONC      CKWok   Ok, il n'est pas déjà ouvert
0648 00342      *
0649 00342 0C      CKWopn P=P+1
0650 00344 0C      CKWsec P=P+1

```

```

0701 00376      *
0702 00376      *
0703 00376      *
0704 00376      *
0705 00376      *
0706 00376      *
0707 00376      *
0708 00376      *
0709 00376      *
0710 00376 05     CHKOPN B=C      A      Sauver l'adresse dans B(A)
0711 00378      *
0712 00378      *
0713 00378      *
0714 00378 32000   LC(3)   =bFIB
0715 00370 8F00000 GOSBVL =/OFND Abime A, C(A), C(S), D1
0716 00384      *
0717 00384      *
0718 00384      *
0719 00384      *
0720 00384      *
0721 00384      *
0722 00384      *
0723 00384      *
0724 00384      *
0725 00384      *
0726 00384 3652000 LCHEX  F00025 C(G-5) := F0 (fichier en Ram)
0727 00380 09      *
0728 00380 09      *
0729 00380 09      *
0730 00380 09      *
0731 00380 09      *
0732 00380 26     P=      6
0733 00391 148     A=DAT1  B
0734 00394 368     ?A=0    B
0735 00397 32     GOYES  CHK090 On n'a pas trouvé ! C'est donc ok
0736 00399 170     D1=D1+ (=oFBEGB)- (=oFILAb)
0737 0039C 170     D1=D1+ (=oDBEGB)- (=oFBEGB)
0738 0039F 1586    A=DAT1  7
0739 003A3 912     ?A=C    WP
0740 003A6 00     GOYES  CHK080 On a trouvé ! Beep ! "File Open"
0741 003A8 170     D1=D1+ (=oRECLb)- (=oDBEGB)
0742 003AB 170     D1=D1+ (=oRELn)- (=oRECLb)
0743 003AE 170     D1=D1+ (=IFIB)- (=oRELn)
0744 003B1 5FD     GONC   CHK010 B.E.T.
0745 003B4      *
0746 003B4 20     CHK080 P=      0
0747 003B6      *
0748 003B6      *
0749 003B6      *

```

```

0651 00346 0C      CKWrom P=P+1
0652 00348 80CF    CKWok  C=P      15
0653 0034C 20      P=      0
0654 0034E 01      RTM
0655 00350      *
0656 00350      *
0657 00350      *
0658 00350      *
0659 00350      *
0660 00350      *
0661 00350      *
0662 00350      *
0663 00350      *
0664 00350      *
0665 00350      *
0666 00350      *
0667 00350      *
0668 00350      *
0669 00350      *
0670 00350      *
0671 00350      *
0672 00350      *
0673 00350      *
0674 00350      *
0675 00350      *
0676 00350      *
0677 00350 A4E    =WRITE? C=C-1  S
0678 00353 400     RTNC      pas d'erreur
0679 00356      *
0680 00356 02      C=0      A
0681 00358 A4E     C=C-1    S
0682 00358 441     GDC      WRTrom
0683 0035C A4E     C=C-1    S
0684 00361 480     GDC      WRTsec
0685 00364 3100    WRTopn LC(2) =oFOPEN
0686 00368 03      RTNCC
0687 0036A 3100    WRTsec LC(2) =oFPROT
0688 0036E 03      RTNCC
0689 00370 3100    WRTrom LC(2) =oFACCS
0690 00374 03      RTNCC
0691 00376      *
0692 00376      *
0693 00376      *
0694 00376      *
0695 00376      *
0696 00376      *
0697 00376      *
0698 00376      *
0699 00376      *
0700 00376      *

```

```

0750 00386 09      C=B      A      On remet les choses en place
0751 00388 02      RTNCC      Cy = 1
0752 0038A      *
0753 0038A 20     CHK090 P=      0
0754 0038C      *
0755 0038C      *
0756 0038C      *
0757 0038C 09     C=B      A      On remet les choses en place
0758 0038E 03     RTNCC
0759 003C0      *
0760 003C0      *
END

```

```

*ATNCLR      Extrn Ukn   - 0035
*BSEERR      Extrn Ukn   - 0462
  bserr      00296 Rel 0551 - 0542 0586
*CHKTXT      00199 Rel 0391 -
*CHKWRT      00318 Rel 0625 -
*CKSREQ      Extrn Ukn   - 0032
*CONVUC      Extrn Ukn   - 0066 0355
*COPYu       Extrn Ukn   - 0540
*CRETF*      Extrn Ukn   - 0584
  CHK010     00391 Rel 0733 - 0744
  CHK030     00384 Rel 0746 - 0740
  CHK090     0038A Rel 0753 - 0735
  CHKOPM     00376 Rel 0710 - 0646
  CKXTX1     00183 Rel 0404 - 0397
  CKW010     00338 Rel 0642 - 0640
  CKWok      00348 Rel 0652 - 0647
  CKWopn     00342 Rel 0649 -
  CKWrom     00346 Rel 0651 - 0631
  CKWsec     00344 Rel 0650 - 0641
*DB*AVS      Extrn Ukn   - 0079
*EXFIL       Extrn Ukn   - 0130
  FILE1      00188 Rel 0427 -
  FILE2      00203 Rel 0485 -
  FINDF      Extrn Ukn   - 0561
  FLIMSK     Extrn Ukn   - 0178
  FLCREG     Extrn Ukn   - 0175
  FPOLL      Extrn Ukn   - 0073
  FSPECx     Extrn Ukn   - 0437
  FL2-10     00240 Rel 0517 - 0508
  FL2-20     0029A Rel 0556 - 0496 0499 0503
  FL2-30     002C6 Rel 0579 - 0569
  FL2-40     00319 Rel 0604 - 0549 0564
  FILNd      003C0 Rel 0760 -
*GETPR1      Extrn Ukn   - 0636
  I/OFND     Extrn Ukn   - 0715
  ISRAM?     Extrn Ukn   - 0629
*KEYNAM      Extrn Ukn   - 0347
*KEYWT       00000 Rel 0030 - 0033
*MGOSUB      Extrn Ukn   - 0539
*MVHEM*      Extrn Ukn   - 0150
*POPBUF      Extrn Ukn   - 0034
*POPUPD      Extrn Ukn   - 0544
*PSHUPD      Extrn Ukn   - 0537
*PURGE       0007E Rel 0130 -
  S-R0-0     Extrn Ukn   - 0431 0441
  SALLOC     Extrn Ukn   - 0519
  SCRLLR     Extrn Ukn   - 0030
  SINTRO     Extrn Ukn   - 0453 0489 0556 0579
  SINTR1     Extrn Ukn   - 0588
  SVINF*     Extrn Ukn   - 0523
  
```

```

Object : obj/xutil.a0
Listing : list/xutil.a1
Date : Sat Aug 12 17:41:27 1989
Errors : 000
  
```

Areuh Assembler/Linker V2.4, (c) P. David & J. Taillandier 1986 Paris, France

```

*SVINFO      Extrn Ukn   - 0528
*TBMSG4      Extrn Ukn   - 0081
  URITe?     00350 Rel 0677 -
  URToop     00354 Rel 0695 -
  URTron     00370 Rel 0689 - 0682
  URtsec     0036A Rel 0687 - 0684
  YNO        00162 Rel 0347 -
  YNO500     00193 Rel 0367 - 0353 0358 0363
  bFIB       Extrn Ukn   - 0714
  bserr      001FC Rel 0462 - 0447 0551
  *eFACCS    Extrn Ukn   - 0589
  *eFOPEN    Extrn Ukn   - 0625
  *eFPROT    Extrn Ukn   - 0687
  *eFSPEC    Extrn Ukn   - 0461 0511
  *eFTYPE    Extrn Ukn   - 0401
  *extsk     00008 Rel 0212 -
  *extsk*    000E8 Rel 0224 -
  fspace     001FG Rel 0460 - 0449
  id         Extrn Ukn   - 0067 0356
  kut10      00015 Rel 0034 - 0031
  IFIB       Extrn Ukn   - 0743
  *oDBEGb    Extrn Ukn   - 0737 0741
  *oFBEGb    Extrn Ukn   - 0736 0737
  *oFIL#b    Extrn Ukn   - 0736
  *oRECLb    Extrn Ukn   - 0741 0742
  *oRELnB    Extrn Ukn   - 0742 0743
  *pTRANS    Extrn Ukn   - 0074
  *poscmd    00027 Rel 0066 -
  *posmg     00034 Rel 0068 - 0357
  posm10     0005F Rel 0100 - 0108
  posm20     0006C Rel 0105 - 0092
  posm50     0007A Rel 0111 - 0102
  *sCREAT    Extrn Ukn   - 0134
  *setf11    000AC Rel 0175 -
  *setf20    000CB Rel 0185 - 0182
  *setf30    000D2 Rel 0188 - 0184
  ska100     00112 Rel 0273 - 0315
  ska200     00154 Rel 0313 - 0271
  ska300     0015C Rel 0322 - 0277 0293
  ska310     0015E Rel 0323 - 0314
  *teKEYS    Extrn Ukn   - 0356
  *teVMD     Extrn Ukn   - 0067
  
```

Source : xutil.as

```

0001 00000 TITLE Utilitaires empruntés à JPC Rom (jpcutil.as)
0002 00000
0003 00000
0004 00000
0005 00000
0006 00000
0007 00000
0008 00000
0009 00000
0010 00000
0011 00000
0012 00000
0013 00000
0014 00000
0015 00000
0016 00000
0017 00000
0018 00000
0019 00000
0020 00000
0021 00000
0022 00000
0023 00000
0024 00000
0025 00000
0026 00000
0027 00000
0028 00000
0029 00000
0030 00000
0031 00000
0032 00000
0033 00000
0034 00000
0035 00000
0036 00000
0037 00000
0038 00000
0039 00000
0040 00000
0041 00000
0042 00000
0043 00000
0044 00000
0045 00000
0046 00000
0047 00000
0048 00000
0049 00000
0050 00000

```

- Num2D1
- But: traduire un nombre en décimal et transférer sa représentation alphanumérique dans le buffer pointé par DI.
- Entrée:
 - - Nombre en hexa dans A(A)
 - - DI pointe sur le premier caractère à renvoyer
 - - P = 0
- Sortie:
 - - DI pointe après le dernier caractère envoyé
 - - mode = hex
 - - P = 0
 - - Cy = 1
 - - C(S) = nombre de chiffres affichés - 1
- Abime: A(A), B(A), C(A), DI
- Appelle: HEXDEC
- Niveaux: 1
- Historique:
 - 86/07/31: supprimer la boucle infinie quand A(A) = 0
 - 86/08/01: ajout de documentation
 - 88/05/23: PD/JT pompage dans KA pour T/XEDIT

```

=Num2D1 ?A=0 A Nb dans A(A), P=0
GOYES num15
COSBVL =HEXDEC Résultat dans A,B,C
SETHEX Car mode = DEC après HEXDEC
• Boucle de comptage du nombre de chiffres
P= 7 Max=1048575
num10 P=P-1
?A=0 P
GOYES num10
• num15: P = nombre de chiffres à envoyer - 1
num15 C=P 15 C(S) := nb chiffres à afficher - 1
• préparation de A(W) pour la boucle d'envoi
num20 ASRC
P=P-1
?P= #P
GOYES num20
P= 0
• Fin de la boucle. A(W) := C1 C2 ... Cn 0 ... 0
LCASC '0' Pour ajouter à tous les chiffres C1
P=C 15 P := nombre de chiffres à envoyer
• Boucle d'envoi des chiffres
num30 ASLC A(W) := C1+1 ... Cn 0 ... 0 C1
A=A+C B A(B) := valeur Ascii du chiffre

```

```

FILMd 0008A Rel 0093 -
=HEXDEC Extrn Ukn - 0030
=KEYBUF Extrn Ukn - 0076 0083
=MOVE=M Extrn Ukn - 0081
=Num2D1 00000 Rel 0028 -
=fkey 00046 Rel 0073 -
fkey10 00084 Rel 0030 - 0088
num10 00010 Rel 0034 - 0036
num15 00017 Rel 0035 - 0029
num20 00018 Rel 0040 - 0043
num30 0002F Rel 0049 - 0055

```

Source : jpcutil.as

Object : obj/jpcutil.as

Listing : list/jpcutil.al

Date : Sat Aug 12 17:41:28 1989

Errors : 000

Areuh Assembler/Linker V2.4. (c) P. David & J. Taillandier 1986 Paris, France

```

0051 00035 149 DAT1=A B envoi dans le buffer
0052 00038 171 DI=DI+ 2
0053 00038 D0 A Pour le chiffre suivant
0054 0003D 00 P=P-1
0055 0003F 5FC GONC num30
0056 00042 20 P= 0 En sortie: A(W)=0, P=0, Cy=1
0057 00044 01 RTN

```

- fkey
- But: mettre une touche en premier dans le key buffer.
- Entrée:
 - - R3(B) = code physique de la touche (cf 10S2, p 12-13)
 - Abime: A(A), B(A), C(A), D0
 - Niveaux: 0
 - Detail: INTOM et INTOFF sont utilisés ici
- Historique:
 - 86/08/01: ajout de documentation
 - 88/05/23: PD/JT pompage dans KA pour T/XEDIT

```

=fkey C=0 A Entrée: R3(B)=physical key-code
LC(2) 28 Nb de quartets du KEYBUF déplacer
B=C A > Nombre de quartets à déplacer
LC(5) =KEYBUF
A=C A > Adresse origine
C=C+1 A
C=C+1 A > Adresse destination
INTOFF
COSBVL =MOVE=M Déplacer 14 entrées du KEYBUF
C=R3 La touche à insérer
D0=(5) =KEYBUF
DAT0=C B Insertion effectivement réalisée
D0=D0- 1 D0=(5) KEYPTR
C=DAT0 S
C=C+1 S Une touche de plus si possible
GOC fkey10
DAT0=C S
fkey10 INTOM
RTN
END

```