```
1    * PeerSoft v1.5.6 by Benoit Gilon - (c) 2006-2015 L.P.C.B.
2    * 30 Sep 2012: initial release
3    * 16 Oct 2012: 1.1, integ. divide support
4    * 30 Dec 2012: 1.2, integer arithmetic in FOR/NEXT loops
5    * & @ pseudo var)
6    * 3nd Jan 2013: 1.3 reorg subroutine #0
7    * 27 Jan 2013: 1.4 reorg subroutine #4 and MT kernel
8    * 6 Apr 2013: local error handling within MT kernel
9    * 1.5.5 addons:
10   * 31st July 2015: can concurrenlty define up to 11
11   * assembly language functions.. support for up to 2
12   * arguments instead of one originally.
13   * 3nd August 2015: support for Procedural functions
14   * 1.5.6 addons:
15   * 8th September 2015: byte new integer subtype added
16   * ToDo: Two new integer subtypes: 24 and
17   * 32 bits integer now understood (convenience for array
18   * variables of this integer subtypes).
19   * ToDo: Possibility to store indiv. array content
20   * within aux mem (auxiliary memory Apple and AE RAMWorks
21   * protocol)
22   * Merlin 8 assembler
26   * Constants
27   VERSION  =     $15
28   K6502    =     0
29   K65C02   =     1
30   K65816   =     1
31   * Generate either 65(816!C)02 compatible version
32   KOPT     =     K65C02
33   KNEW     =     1
34   KNEW2    =     1
35   KOPTLNG32 =    1
36   KOPTLNG33 =    0
37   * Cache size (# of entries) for simple variables
38   KSNCACH  =     4
39   * Cache size (# of entries) for array variables
40   KANCACH  =     4
41
43            XC
44   KOPT16   =     1
47            XC
52
53   * Token equates
54   TOKEQUAL =     $D0
55   TOKADD   =     $C8
56   TOKMUL   =     $CA
57   TOKDIV   =     $CB
58   TOKDEF   =     $B8        Prefix for DEF(INT!STR!SNG)
59   TOKINT   =     $D3        DEFINT instr st. as 2 tokens
60   TOKUSR   =     $D5        DEFUSR...
61   TOKMINUS =     $C9
62   TOKREM   =     $B2
63   TOKDATA  =     $83
64   TOKIF    =     $AD
65   TOKFN    =     $C2
66   TOKTO    =     $C1
67   TOKSTRD  =     $E4
68   TOKCHRD  =     $E7
69   TOKSGN   =     $D2
```

```
70   TOKSCRN  =     $D7
71   TOKNOT   =     $C6
72   TOKSTEP  =     $C7
73   TOKGOSUB =     $B0
74   TOKGOTO  =     $AB
75   TOKFOR   =     $81
76
77   * Page zero and monitor equates
78   PCL      EQU   $3A
79   LENGTH   EQU   $2F
80   INSDS2   EQU   $F88C
81   PCADJ    EQU   $F953
82   A1L      EQU   $3C
83   A2L      EQU   $3E
84   A4L      EQU   $42
85   MOVE     EQU   $FE2C
86   CH       EQU   $24
87   XFER     EQU   $C314
88   VECZAUX  EQU   $03ED
89
90   * Applesoft equates
91   DIMFLG   EQU   $10        Input to PTRGET
92   * Output from PTRGET
93   VALTYP   EQU   $11        $FF if string, 0 if num.
94   INTTYP   EQU   $12        $80 if integer, 0 otherwise
95   VARNAM   EQU   $81        Encoded varname 1st char.
96   VARPNT   EQU   $83        Variable value pointer
97   SUBFLG   EQU   $14        Parameter for PTRGET routine
98   LINNUM   EQU   $50        Line # (output from LINGET)
99   CURLIN   EQU   $75        Current line # (being run)
100  INDEX    EQU   $5E        General ptr for ROM str. routines
101  LOWTR    EQU   $9B        Address of BASIC line (output fro
m FNDLIN)
102  FAC      EQU   $9D        Main floating point accumulator
103  DEST     EQU   $60        Used by NEXT
104  STREND   EQU   $6D        End of array memory
105  FACSIGN  EQU   $A2
106  FACLO    EQU   $A1
107  FACMO    EQU   $A0
108  TXTPTR   EQU   $B8        Pointer to BASIC program memory
109  OLDTPTR  EQU   $79
110  REMSTK   EQU   $F8
111  OLDTEXT  EQU   $79
112  ARYPNT   EQU   $94        Pointer to array structure
113  ERRFLG   EQU   $D8        ONERR activivty flag
114  ERRLIN   EQU   $DA        Offending line #
115  ERRPOS   EQU   $DC        Where in the offending line #..
116  ERRNUM   EQU   $DE        Error #
117  ERRSTK   EQU   $DF        Stack pntr of offending instr.
118  TXTPSV   EQU   $F4
119  CURLSV   EQU   $F6
120
121  TOKTABL  EQU   $D0D0      Address of internal Applesoft tok
en table
122  ISLETC   EQU   $E07D      Check whether current char alpha
123  SYNERR   EQU   $DEC9      Report a SYNTAX ERROR
124  VLET     EQU   $DA46
```

```
                125  VPTRGET  EQU   $DFEF      PTRGET return adress (from stack)
                126  ISCNTC   EQU   $D858      Check for Ctrl-C keystroke
                127  ADDON    EQU   $D998      Add Y to TXTPTR
                128  LINGET   EQU   $DA0C      Get line number from TXTPTR
                129  CHKMEM   EQU   $D3D6      Check for A 16bit words on stack
                130  COMBYTE  EQU   $E74C      Check for comma and compute
                131
                132  * Applesoft output routines
                133  OUTDO    EQU   $DB5C      Generic
                134  CRDO     EQU   $DAFB      Carriage return
                135  OUTSPC   EQU   $DB57      Space
                136  FNDLIN   EQU   $D61A      From line number (LINNUM) to addr
ess
                137  NEWSTT   EQU   $D7D2      Applesoft main exec loop
                138  FORPNT   EQU   $85
                139  FRMEVL   EQU   $DD7B      Eval. expr pointed to by TXTPTR
                140  FRMNUM   EQU   $DD67      Eval. expr & ensure numeric resul
t
                141  GETADR   EQU   $E752      Expression to 16bits integer
                142  GETBYT   EQU   $E6F8      Eval. expr into single byte value
                143  * Some checking about FAC:must contain..
                144  CHKNUM   EQU   $DD6A      a scalar factor
                145  CHKSTR   EQU   $DD6C      a string factor
                146  AYINT    EQU   $E10C      Integer conversion from FP
                147  * Some floating point computing dst is FAC1
                148  FSUB     EQU   $E7A7      (Y,A) - FAC1
                149  FADD     EQU   $E7BE      (Y,A) + FAC1
                150  FMULT    EQU   $E97F      (Y,A) * FAC1
                151  FDIV     EQU   $EA66      (Y,A) / FAC1
                152  NEGOP    EQU   $EED0      -FAC1
                153  * Raise some Applesoft errors
                154  GOSTLERR EQU   $E5B2      STRING TOO LONG
                155  GOOVFERR EQU   $E8D5      OVERFLOW
                156  GOTMIERR EQU   $DD76      TYPE MISMATCH
                157  GODVZERR EQU   $EAE1      DIVIDE BY ZERO
                158  GOIQERR  EQU   $E199      ILLEGAL QUANTITY
                159  FREESPC  EQU   $71
                160  STRSPA   EQU   $E3DD      Get space from string pool for a
string of len A
                161  DSCTMP   EQU   $9D        Temporary string pointer
                162  STRING1  EQU   $AB        String pointer used by copy
                163  MOVINS   EQU   $E5D4      Move string(STRING1) into memory(
FRESPC)
                164  ERRDIR   EQU   $E306      Raises a illegal direct mode iif
required
                165  DATAN    EQU   $D9A3      Scan ahead to next EOI
                166  DATA     EQU   $D995      TXTPTR points to next separator
                167  VARTAB   EQU   $69        Begin of simple var. mem. area
                168  ARYTAB   EQU   $6B        Begin of array var. mem. area
                169
                170  FRMSTCK3 EQU   $DE20
                171
                172  * ZP slots used by integer signed 16bits mult/div subroutin
es
                173  MCAND    EQU   $C0
                174  MPLIER   EQU   $C2
                175  DIVEND   EQU   MPLIER
```

```
176  DIVSOR   EQU   $C0
177  PARTIAL  EQU   $BE
178  AUXBANK  EQU   $BF
179  LETINF   EQU   $C0
180  TYPMOD   EQU   $C1
181  INTTYPSV EQU   $C7
182  VALTYPSV EQU   $C8
183
184  * DOS 3.3 equates
185  OPRND    EQU   $44
186  DBUFP    EQU   $9D00
187
188           ORG   $4000
189
190  AUXPTR   EQU   $06
191  IDMOCL   EQU   $BD
192  OFFSET   EQU   $C2
193  XSAV     EQU   $B4
194  YSAV     EQU   $B5
195  MODREM   EQU   $BE
196  MODDAT   EQU   $BF
197  GFLAG    EQU   $C0
198  IDX0     EQU   $C0
199  DEFFLG   EQU   $C1
200  NOPER    =     4
201
204  EMOV     MAC
205           LDA   ]1
206           STA   ]2
207           <<<
208
209  STD      MAC
210           EMOV  ]1;]2
211           EMOV  ]1+1;]2+1
212           <<<
213
214  * 16bits immediate store
215  STID     MAC
216           EMOV  #]1;]2
217           EMOV  #>]1;]2+1
218           <<<
219
220  * Copy a large memory area within
221  * adressable memory
222  MOVM     MAC
223           STID  ]1;A1L
224           STID  ]2;A2L
225           STID  ]3;A4L
226           JSR   MOVE
227           <<<
228
229  * Copy a small memory area within
230  * adressable memory
231  SMOVE    MAC
232           LDX   #]3
233  LOOP     LDA   ]1-1,X
234           STA   ]2-1,X
```

```
235           DEX
236           BNE   LOOP
237           <<<
238
239   * Macros for simulating 65C02 instructions
240   * on a 6502
241   MPHX    MAC
242           DO    KOPT-K65C02
243           TXA
244           PHA
245           ELSE
246           PHX
247           FIN
248           <<<
249
250   MPHY    MAC
251           DO    KOPT-K65C02
252           TYA
253           PHA
254           ELSE
255           PHY
256           FIN
257           <<<
258
259   MPLX    MAC
260           DO    KOPT-K65C02
261           PLA
262           TAX
263           ELSE
264           PLX
265           FIN
266           <<<
267
268   MPLY    MAC
269           DO    KOPT-K65C02
270           PLA
271           TAY
272           ELSE
273           PLY
274           FIN
275           <<<
276
277   MTSB    MAC
278           DO    KOPT-K65C02
279           ORA   ]1
280           STA   ]1
281           ELSE
282           TSB   ]1
283           FIN
284           <<<
285
286   GOTO    MAC
287           DO    KOPT-K6502
288           BRA   ]1
289           ELSE
290           JMP   ]1
291           FIN
```

```
                292           <<<
                294
                295   * Do all the stuff for installing Peersoft
                296   * between DOS and its buffers
                297           PUT   PEERINSTALL
               >1     NEWY     EQU   $47
               >3     * For the Smart disassembler technology introduced
               >4     * in Peersoft v1.5 for 65816 systems
               >5             DUMMY A1L
003C: 00       >6     FC       DS    1
003D: 00       >7     FE       DS    1
003E: 00       >8     FM       DS    1
003F: 00       >9     FX       DS    1
0040: 00       >10    FCSTK    DS    1
0041: 00       >11    FXSTK    DS    1
0042: 00       >12    FMSTK    DS    1
               >13            DEND
               >15
               >16    * This module deals with all installation stuff for the
               >17    * Peersoft suite
4000: A9 D3    >18    SUITE    LDA   #$9CD3     Compute the offset
4002: 38       >19             SEC              ;Put it in :0+1 (lobyte)
4003: ED 00 9D >20             SBC   DBUFP       and :1+1 (hibyte)
4006: 8D 4F 40 >21             STA   :0+1
4009: A9 9C    >22             LDA   #>$9CD3
400B: ED 01 9D >23             SBC   DBUFP+1
400E: AA       >24             TAX
400F: 0D 4F 40 >25             ORA   :0+1
               >26    * If first utility to ask for memory this way, then ask for
               >27    * one additional page for our own purpose (i.e. Bananasoft
               >28    * or Peersoft)
4012: F0 01    >29             BEQ   :6
4014: CA       >30             DEX
4015: 8E 57 40 >31    :6       STX   :1+1
               >32
               >33    * Relocate code (don´t move it yet)
4018: A9 01    >35             LDA   #1
401A: 85 3D    >36             STA   FE
401C: 85 3F    >37             STA   FX
401E: 85 3E    >38             STA   FM
4020: A9 CF    >40             LDA   #AROMBA
4022: A0 47    >41             LDY   #>AROMBA
4024: 85 3A    >42    ]LOOP    STA   PCL
4026: C9
Unknown label in line: 297 >43
               >43             CMP   #FCODE-FNDVAR2+AROMBA
4029: 98       >44             TYA
402A: E9
Unknown label in line: 297 >45
               >45             SBC   #>FCODE-FNDVAR2+AROMBA
402D: B0 31    >46             BCS   :4
402F: 84 3B    >47             STY   PCL+1
4031: 20 27 42 >51             JSR   MINSDS2
4034: A4 2F    >52             LDY   LENGTH
4036: C0 02    >53             CPY   #2         Only relocates 3 bytes instr.
4038: D0 20    >54             BNE   :3
403A: B1 3A    >55             LDA   (PCL),Y
```

```
403C: AA       >56            TAX
403D: 88       >57            DEY
403E: B1 3A    >58            LDA   (PCL),Y
4040: A8       >59            TAY
4041: C9
Unknown label in line: 297 >60
               >60            CMP   #FIN       Only if adress within range
4044: 8A       >61            TXA
4045: E9
Unknown label in line: 297 >62
               >62            SBC   #>FIN
4048: B0 10    >63            BCS   :3         Must be < FIN to be relocated
404A: C0 A6    >64            CPY   #FNDVAR2
404C: 8A       >65            TXA
404D: E9 7A    >66            SBC   #>FNDVAR2
404F: 90 09    >67            BCC   :3         Must be >= FNDVAR2
4051: 98       >68            TYA              ;Relocates address
4052: E9 00    >69   :0       SBC   #0
4054: A0 01    >70            LDY   #1
4056: 91 3A    >71            STA   (PCL),Y    Low byte
4058: C8       >72            INY
4059: 8A       >73            TXA
405A: E9 00    >74   :1       SBC   #0
405C: 91 3A    >75            STA   (PCL),Y    High byte
405E: 20 53 F9 >76   :3       JSR   PCADJ      Adjust PCL to length byte
4061: 4C 24 40 >77            JMP   ]LOOP      Loop
               >78
               >80
               >81   * Relocate some non trivial references (i.e. instructions
               >82   * with immediate adressing mode).
4064: A2
Unknown label in line: 297 >83
               >83   :4       LDX   #ADPFT-ADPFB-1

Unknown label in line: 297 >84
               >84   ]LOOP    LDA   ADPFB+AROMBA-FNDVAR2,X
4069: 38       >85            SEC
406A: ED 4F 40 >86            SBC   :0+1

Unknown label in line: 297 >87
               >87            STA   ADPFB+AROMBA-FNDVAR2,X

Unknown label in line: 297 >88
               >88            LDA   ADPFT+AROMBA-FNDVAR2,X
4071: ED 57 40 >89            SBC   :1+1

Unknown label in line: 297 >90
               >90            STA   ADPFT+AROMBA-FNDVAR2,X
4076: CA       >91            DEX
4077: 10 EE    >92            BPL   ]LOOP
               >93
4079: A2 0B    >94            LDX   #ADT1-ADB1-1
407B: A9 00    >95            LDA   #0
407D: 85 3A    >96            STA   PCL
407F: BD D4 42 >97   ]LOOP    LDA   ADT1,X
4082: 85 3B    >98            STA   PCL+1
4084: BC C8 42 >99            LDY   ADB1,X
```

```
4087: B1 3A    >100           LDA   (PCL),Y
4089: 38       >101           SEC
408A: ED 4F 40 >102           SBC   :0+1
408D: 91 3A    >103           STA   (PCL),Y
408F: BD EC 42 >104           LDA   ADT2,X
4092: 85 3B    >105           STA   PCL+1
4094: BC E0 42 >106           LDY   ADB2,X
4097: B1 3A    >107           LDA   (PCL),Y
4099: ED 57 40 >108           SBC   :1+1
409C: 91 3A    >109           STA   (PCL),Y
409E: CA       >110           DEX
409F: 10 DE    >111           BPL   ]LOOP
               >112
40A1: A2
Unknown label in line: 297 >113
               >113           LDX   #OFFSTT-OFFSTB-1

Unknown label in line: 297 >114
               >114  ]LOOP    LDA   OFFSTB+AROMBA-FNDVAR2,X
40A6: 38       >115           SEC
40A7: ED 4F 40 >116           SBC   :0+1

Unknown label in line: 297 >117
               >117           STA   OFFSTB+AROMBA-FNDVAR2,X

Unknown label in line: 297 >118
               >118           LDA   OFFSTT+AROMBA-FNDVAR2,X
40AE: ED 57 40 >119           SBC   :1+1

Unknown label in line: 297 >120
               >120           STA   OFFSTT+AROMBA-FNDVAR2,X
40B3: CA       >121           DEX
40B4: 10 EE    >122           BPL   ]LOOP
               >123  * Move the code
40B6: A9 A6    >124           LDA   #CGARBAG
40B8: A2 7A    >125           LDX   #>CGARBAG
40BA: 38       >126           SEC
40BB: ED 4F 40 >127           SBC   :0+1
40BE: 85 42    >128           STA   A4L
40C0: 8A       >129           TXA
40C1: ED 57 40 >130           SBC   :1+1
40C4: 85 43    >131           STA   A4L+1
               >132
40C6: A9 CF    >133           LDA   #CGARBAG+AROMBA-FNDVAR2
40C8: A2 47    >134           LDX   #>CGARBAG+AROMBA-FNDVAR2
40CA: 85 3C    >135           STA   A1L
40CC: 86 3D    >136           STX   A1L+1
               >137
40CE: A9
Unknown label in line: 297 >138
               >138           LDA   #FIN-1+AROMBA-FNDVAR2
40D1: 85 3E    >138           STA   A2L
40D3: A9
Unknown label in line: 297 >138
               >138           LDA   #>FIN-1+AROMBA-FNDVAR2
40D6: 85 3F    >138           STA   A2L+1
               >139
```

```
40D8: A0 00    >140           LDY   #0
40DA: 2C 81 C0 >141           BIT   $C081
40DD: 2C 81 C0 >142           BIT   $C081
40E0: 20 2C FE >143           JSR   MOVE
               >144  * Reconstruct DOS buffers below PeerSoft
40E3: AD 00 9D >145           LDA   DBUFP
40E6: AE 01 9D >146           LDX   DBUFP+1
40E9: C9 D3    >147           CMP   #$9CD3
40EB: D0 05    >148           BNE   :7
40ED: E0 9C    >149           CPX   #>$9CD3
40EF: D0 01    >150           BNE   :7         One more page if first utility
40F1: CA       >151           DEX              ; to install this way
40F2: 38       >152  :7       SEC
40F3: E9
Unknown label in line: 297 >153
               >153           SBC   #LONGLANG
40F6: A8       >154           TAY
40F7: 8A       >155           TXA
40F8: E9
Unknown label in line: 297 >156
               >156           SBC   #>LONGLANG
40FB: 8C 00 9D >157           STY   DBUFP      New DOS base buffer address
40FE: 8D 01 9D >158           STA   DBUFP+1
4101: 20 D4 A7 >159           JSR   $A7D4
               >160
4104: A9 15    >161           LDA   #VERSION

Unknown label in line: 297 >162
               >162           STA   PVERSION
4108: A9 80    >163           LDA   #$80

Unknown label in line: 297 >164
               >164           STA   OPTCGOTO

Unknown label in line: 297 >166
               >166           STZ   NEEDDEC
               >171
               >172  * Number of Applesoft instruction runs
               >173  * between two consecutives context switches
410E: A9 0A    >174           LDA   #10

Unknown label in line: 297 >175
               >175           STA   ICTRACTV

Unknown label in line: 297 >180
               >180           STZ   MTACTV
4114: A9 4C    >182           LDA   #$4C

Unknown label in line: 297 >183
               >183           STA   REVECTOR

Unknown label in line: 297 >184
               >184           STA   VGARBAG
411A: 38       >185           SEC
411B: A9 4C    >186           LDA   #ROUTGEN
411D: ED 4F 40 >187           SBC   :0+1
```

```
Unknown label in line: 297 >188
                >188            STA   REVECTOR+1
4122: A9 8B     >189            LDA   #>ROUTGEN
4124: ED 57 40 >190             SBC   :1+1

Unknown label in line: 297 >191
                >191            STA   REVECTOR+2
4129: A9 FA     >192            LDA   #NPTRGL90
412B: ED 4F 40 >193             SBC   :0+1

Unknown label in line: 297 >194
                >194            STA   VNPTRG90
4130: A9 7E     >195            LDA   #>NPTRGL90
4132: ED 57 40 >196             SBC   :1+1

Unknown label in line: 297 >197
                >197            STA   VNPTRG90+1
4137: A9 E2     >198            LDA   #NARRGL91
4139: ED 4F 40 >199             SBC   :0+1

Unknown label in line: 297 >200
                >200            STA   VNARRG91
413E: A9 7F     >201            LDA   #>NARRGL91
4140: ED 57 40 >202             SBC   :1+1

Unknown label in line: 297 >203
                >203            STA   VNARRG91+1
4145: A9
Unknown label in line: 297 >204
                >204            LDA   #TABOFB
4148: ED 4F 40 >205             SBC   :0+1

Unknown label in line: 297 >206
                >206            STA   ADADR
414D: A9
Unknown label in line: 297 >207
                >207            LDA   #>TABOFB
4150: ED 57 40 >208             SBC   :1+1

Unknown label in line: 297 >209
                >209            STA   ADADR+1
4155: A2 84     >210            LDX   #GARBAG
4157: A9 E4     >211            LDA   #>GARBAG

Unknown label in line: 297 >212
                >212            BIT   MEMORY
415B: 10 0B     >213            BPL   *+13
415D: A9 3C     >214            LDA   #NGARBAG
415F: ED 4F 40 >215             SBC   :0+1
4162: AA        >216            TAX
4163: A9 82     >217            LDA   #>NGARBAG
4165: ED 57 40 >218             SBC   :1+1

Unknown label in line: 297 >219
                >219            STX   VGARBAG+1

Unknown label in line: 297 >220
```

```
                >220            STA   VGARBAG+2
416C: A9
Unknown label in line: 297 >221
                >221            LDA   #NDSVCMD   New DOS Save for applesoft
416F: ED 4F 40  >222            SBC   :0+1
4172: 8D A6 A3  >223            STA   $A3A6
4175: A9
Unknown label in line: 297 >224
                >224            LDA   #>NDSVCMD
4178: ED 57 40  >225            SBC   :1+1
417B: 8D A7 A3  >226            STA   $A3A7
417E: A9
Unknown label in line: 297 >227
                >227            LDA   #NDLVCMD   Part of routine for loading
4181: ED 4F 40  >228            SBC   :0+1
4184: 8D 2E A4  >229            STA   $A42E
4187: A9
Unknown label in line: 297 >230
                >230            LDA   #>NDLVCMD
418A: ED 57 40  >231            SBC   :1+1
418D: 8D 2F A4  >232            STA   $A42F
4190: A9 20     >233            LDA   #$20
4192: 8D 9E 9E  >234            STA   $9E9E
4195: A9
Unknown label in line: 297 >235
                >235            LDA   #NKBDINT
4198: ED 4F 40  >236            SBC   :0+1
419B: 8D 9F 9E  >237            STA   $9E9F
419E: A9
Unknown label in line: 297 >238
                >238            LDA   #>NKBDINT
41A1: ED 57 40  >239            SBC   :1+1
41A4: 8D A0 9E  >240            STA   $9EA0
41A7: 20 F8 42  >241            JSR   BIGRECON
41AA: 20 A2 43  >242            JSR   MOUSEDET

Unknown label in line: 297 >243
                >243            BIT   MEMORY
41AF: 50 12     >244            BVC   :44
                >245  * Copy $F8-$FF pages within ROM to main and aux
                >246  * memory banks
41B1: 20 06 44  >247            JSR   COPYROM
                >248  * Initialize BF page
41B4: 20 C5 44  >249            JSR   INITBF
41B7: 20 DA 41  >250            JSR   MZRTAUX
41BA: 2C 80 C0  >251  :44       BIT   $C080
41BD: 2C 80 C0  >252            BIT   $C080
                >253  * If Applesoft is the active language, so
                >254  * install Peersoft CHRGET/CHRGOT patch
41C0: AD B6 AA  >255  EK        LDA   $AAB6
41C3: F0 12     >256            BEQ   :11
41C5: 2C 81 C0  >257            BIT   $C081
41C8: 2C 81 C0  >258            BIT   $C081
41CB: 20 67 86  >259            JSR   SETUPB
41CE: 4C 7E 86  >260  :11       JMP   SETUPD
                >261
41D1: A9 BF     >262  MZRTAUX   LDA   #$BF
```

```
41D3: A2 00    >263           LDX   #0
41D5: 8D EE 03 >264           STA   $03EE
41D8: 8E ED 03 >265           STX   $03ED
41DB: B8       >266           CLV
41DC: 38       >267           SEC
41DD: 4C 14 C3 >268           JMP   XFER
               >269
               >271  MC       DO    KOPT16
41E0: A3 83 F4 >272           HEX   A383F4D4   LDA d,S/STA d,S/PEA/PEI
41E4: 9B BB    >273           HEX   9BBB       TXY/TYX
41E6: DA FA 04 >275           HEX   DAFA041A3A PHX/PLX/TSB d/INC/DEC
41EB: 7C 80 7A >276           HEX   7C807A5A   JMP (abs, X)/BRA d/PLY/PHY
41EF: 64 9E    >277           HEX   649E       STZ d/STZ a, X
41F1: 0C 9C    >278           HEX   0C9C       TSB a/STZ a
41F3: 1C 14    >279           HEX   1C14       TRB a/TRB d
41F5: B2       >280           HEX   B2         LDA (d)
41F6: 18 38 FB >282  MC1      HEX   1838FB     CLC/SEC/XCE
41F9: 08 28    >283           HEX   0828       PHP/PLP
41FB: C2 E2    >284           HEX   C2E2       REP/SEP
41FD: A2 A0    >285           HEX   A2A0       LDX #/LDY #
               >287  LN       DO    KOPT16
41FF: 01 01 02 >288           HEX   01010201   LDA d,S/STA d,S/PEA/PEI
4203: 00 00    >289           HEX   0000       TXY/TYX
4205: 00 00 01 >291           HEX   0000010000 PHX/PLX/TSB d/INC/DEC
420A: 02 01 00 >292           HEX   02010000   JMP (abs, X)/BRA d/PLY/PHY
420E: 01 02    >293           HEX   0102       STZ d/STZ a, X
4210: 02 02    >294           HEX   0202       TSB a/STZ a
4212: 02 01    >295           HEX   0201       TRB a/TRB d
4214: 01       >296           HEX   01         LDA (d)
4215: 00 00 00 >298           HEX   000000     CLC/SEC/XCE
4218: 00 00    >299           HEX   0000       PHP/PLP
421A: 01 01    >300           HEX   0101       REP/SEP
421C: 01 01    >301           HEX   0101       LDX #/LDY #
               >303  * Check 65C02/65802 used and new machine codes
421E: B2 3A    >304  MINSDS2  LDA   (PCL)
4220: A2 1E    >305           LDX   #LN-MC-1
4222: DD E9 41 >306  ]LOOP    CMP   MC,X
4225: F0 23    >307           BEQ   :0
4227: CA       >308           DEX
4228: 10 F8    >309           BPL   ]LOOP
422A: E8       >310           INX                ;X = 0
               >312  * Grabs all immediate Opcodes with Accumulator
               >313  * eg LDA #, ADC #, ORA # and so on..
422B: A8       >314           TAY
422C: 29 1F    >315           AND   #$1F
422E: C9 09    >316           CMP   #$09
4230: D0 14    >317           BNE   :1
4232: A5 3D    >318           LDA   FE
4234: 05 3E    >319           ORA   FM
4236: D0 0E    >320           BNE   :1
4238: A9 02    >321           LDA   #2
423A: 85 2F    >322           STA   LENGTH
423C: 60       >323           RTS
423D: 98       >324  :1       TYA
423E: 4C 8C F8 >326           JMP   INSDS2
4241: BD 08 42 >327  :0       LDA   LN,X
4244: 85 2F    >328           STA   LENGTH
```

```
4246: E0 16    >330           CPX   #MC1-MC
4248: B0 03    >331           BCS   *+5
424A: A2 00    >332           LDX   #0
424C: 60       >333           RTS
424D: BD A9 42 >334           LDA   OFFX16-MC1+MC,X
4250: 8D 5D 42 >335           STA   OPBASE-1
4253: 80 FE    >336           BRA   *
               >337  OPBASE    EQU   *
4255: 64 3C    >338  OPCLC     STZ   FC
4257: 60       >339           RTS
4258: A9 01    >340  OPSEC     LDA   #1
425A: 85 3C    >341           STA   FC
425C: 60       >342           RTS
425D: A5 3D    >343  OPXCE     LDA   FE
425F: A4 3C    >344           LDY   FC
4261: 85 3C    >345           STA   FC
4263: 84 3D    >346           STY   FE
4265: F0 0D    >347           BEQ   :0
4267: 84 3F    >348           STY   FX
4269: 84 3E    >349           STY   FM
426B: 60       >350  :0        RTS
426C: A5 3C    >351  OPPHP     LDA   FC
426E: 85 40    >352           STA   FCSTK
4270: A5 3F    >353           LDA   FX
4272: 85 41    >354           STA   FXSTK
4274: A5 3E    >355           LDA   FM
4276: 85 42    >356           STA   FMSTK
4278: 60       >357           RTS
4279: A5 40    >358  OPPLP     LDA   FCSTK
427B: 85 3C    >359           STA   FC
427D: A5 3D    >360           LDA   FE
427F: D0 11    >361           BNE   :0
4281: A5 41    >362           LDA   FXSTK
4283: 85 3F    >363           STA   FX
4285: A5 42    >364           LDA   FMSTK
4287: 85 3E    >365           STA   FM
4289: 60       >366  :0        RTS
428A: A2 01    >367  OPSEP     LDX   #1
428C: 2C       >368           HEX   2C          Skip next two bytes
428D: A2 00    >369  OPREP     LDX   #0
428F: A0 01    >370           LDY   #1
4291: B1 3A    >371           LDA   (PCL),Y
4293: A8       >372           TAY
4294: 29 01    >373           AND   #1          bit C involved
4296: F0 0B    >374           BEQ   :0          No
4298: 86 3C    >375           STX   FC
429A: A5 3D    >376  :0        LDA   FE
429C: D0 17    >377           BNE   :1
429E: 98       >378           TYA
429F: 29 20    >379           AND   #$20        bit M involved
42A1: F0 0B    >380           BEQ   :2
42A3: 86 3E    >381           STX   FM
42A5: 98       >382  :2        TYA
42A6: 29 10    >383           AND   #$10        bit X involved
42A8: F0 0B    >384           BEQ   :1
42AA: 86 3F    >385           STX   FX
42AC: 60       >386  :1        RTS
```

```
42AD: A5 3D    >387  OPLDXYI  LDA   FE
42AF: 05 3F    >388           ORA   FX
42B1: D0 0B    >389           BNE   :0
42B3: E6 2F    >390           INC   LENGTH
42B5: 60       >391  :0       RTS
               >398
42B6: 00 03 08 >400  OFFX16   DFB   OPCLC-OPBASE,OPSEC-OPBASE,OPXCE-OPBASE
42B9: 17 24    >401           DFB   OPPHP-OPBASE,OPPLP-OPBASE
42BB: 35 38    >402           DFB   OPSEP-OPBASE,OPREP-OPBASE
42BD: 58 58    >403           DFB   OPLDXYI-OPBASE,OPLDXYI-OPBASE
42BF: D2       >405  ADB1     DFB   EK+9
42C0: D5       >406           DFB   EK+12
42C1: 97       >407           DFB   SETUPB+7+AROMBA-FNDVAR2
42C2: 9F       >408           DFB   SETUPB+15+AROMBA-FNDVAR2
42C3: A8       >409           DFB   SETUPD+1+AROMBA-FNDVAR2

Unknown label in line: 297 >410
               >410           DFB   STP1+1+AROMBA-FNDVAR2

Unknown label in line: 297 >411
               >411           DFB   SFE1+1+AROMBA-FNDVAR2
42C8: BD       >412           DFB   SETLTR+1
42C9: EE       >416           DFB   GN65536+1+AROMBA-FNDVAR2
42CA: E4       >417           DFB   GN32768+1+AROMBA-FNDVAR2
42CB: F3       >418           DFB   GP65536+1+AROMBA-FNDVAR2
42CC: 31       >420           DFB   NAMNTFND+4
42CD: 41       >428  ADT1     DFB   >EK+9
42CE: 41       >429           DFB   >EK+12
42CF: 53       >430           DFB   >SETUPB+7+AROMBA-FNDVAR2
42D0: 53       >431           DFB   >SETUPB+15+AROMBA-FNDVAR2
42D1: 53       >432           DFB   >SETUPD+1+AROMBA-FNDVAR2

Unknown label in line: 297 >433
               >433           DFB   >STP1+1+AROMBA-FNDVAR2

Unknown label in line: 297 >434
               >434           DFB   >SFE1+1+AROMBA-FNDVAR2
42D6: 8D       >435           DFB   >SETLTR+1
42D7: 5F       >439           DFB   >GN65536+1+AROMBA-FNDVAR2
42D8: 5F       >440           DFB   >GN32768+1+AROMBA-FNDVAR2
42D9: 5F       >441           DFB   >GP65536+1+AROMBA-FNDVAR2
42DA: 7F       >443           DFB   >NAMNTFND+4
42DB: D3       >451  ADB2     DFB   EK+10
42DC: D6       >452           DFB   EK+13
42DD: 9B       >453           DFB   SETUPB+11+AROMBA-FNDVAR2
42DE: A3       >454           DFB   SETUPB+19+AROMBA-FNDVAR2
42DF: AD       >455           DFB   SETUPD+6+AROMBA-FNDVAR2

Unknown label in line: 297 >456
               >456           DFB   STP1+3+AROMBA-FNDVAR2

Unknown label in line: 297 >457
               >457           DFB   SFE1+3+AROMBA-FNDVAR2
42E4: C1       >458           DFB   SETLTR+5
42E5: F0       >462           DFB   GN65536+3+AROMBA-FNDVAR2
42E6: E6       >463           DFB   GN32768+3+AROMBA-FNDVAR2
42E7: F5       >464           DFB   GP65536+3+AROMBA-FNDVAR2
```

```
42E8: 32       >466           DFB   NAMNTFND+5
42E9: 41       >474  ADT2     DFB   >EK+10
42EA: 41       >475           DFB   >EK+13
42EB: 53       >476           DFB   >SETUPB+11+AROMBA-FNDVAR2
42EC: 53       >477           DFB   >SETUPB+19+AROMBA-FNDVAR2
42ED: 53       >478           DFB   >SETUPD+6+AROMBA-FNDVAR2

Unknown label in line: 297 >479
               >479           DFB   >STP1+3+AROMBA-FNDVAR2

Unknown label in line: 297 >480
               >480           DFB   >SFE1+3+AROMBA-FNDVAR2
42F2: 8D       >481           DFB   >SETLTR+5
42F3: 5F       >485           DFB   >GN65536+3+AROMBA-FNDVAR2
42F4: 5F       >486           DFB   >GN32768+3+AROMBA-FNDVAR2
42F5: 5F       >487           DFB   >GP65536+3+AROMBA-FNDVAR2
42F6: 7F       >489           DFB   >NAMNTFND+5
               >497
42F7: 2C 81 C0 >498  BIGRECON BIT   $C081
42FA: 2C 81 C0 >499           BIT   $C081
               >500  * What is the model/ROM version of the Apple
42FD: A0 07    >501           LDY   #8-1
42FF: AD B3 FB >502           LDA   $FBB3
4302: 4D C0 FB >503           EOR   $FBC0
4305: 4D BF FB >504           EOR   $FBBF
4308: D9 78 43 >505  ]LOOP    CMP   MACMAT,Y
430B: F0 05    >506           BEQ   :1
430D: 88       >507           DEY
430E: 10 F8    >508           BPL   ]LOOP
4310: C8       >509           INY                ;Assuming default 2+
               >510  * Apple //e enhanced ROM and //gs have same signature,
               >511  * so we ll make the difference on $FC5C
               >512  * value ($EB in a //gs ROM)
4311: C0 02    >513  :1       CPY   #2
4313: D0 21    >514           BNE   :2
4315: AD 5C FC >515           LDA   $FC5C
4318: C9 EB    >516           CMP   #$EB
431A: D0 1A    >517           BNE   :2
431C: A0 08    >518           LDY   #8           //gs!
431E: 18       >519           CLC
431F: FB       >520           HEX   FB           ;XCE: Enter native mode
4320: 08       >521           PHP                ;Push carry status (old emu bit)
4321: C2 30    >522           HEX   C230         Set 16bits mode
4323: 20 1F FE >523           JSR   $FE1F        Call ID firmware routine
4326: 84 47    >524           STY   NEWY
4328: 28       >525           PLP                ;Restore original emulation bit
4329: FB       >526           HEX   FB           ;XCE: Exit native mode
432A: A0 0C    >527           LDY   #12
432C: A5 48    >528           LDA   NEWY+1
432E: D0 06    >529           BNE   :2
4330: A5 47    >530           LDA   NEWY
4332: 09 08    >531           ORA   #8
4334: A8       >532           TAY
               >533
4335: B9 80 43 >534  :2       LDA   MCODE,Y

Unknown label in line: 297 >535
```

```
                >535            STA   MACHINE
433A: 98        >536            TYA
433B: AA        >537            TAX
433C: D0 28     >538            BNE   :3         00 if Apple 2+
                >539  * Test for Apple2+, X=0 upon entry
                >540  * Possible language card being there..
433E: 2C 83 C0  >541            BIT   $C083
4341: 2C 83 C0  >542            BIT   $C083
4344: AD 00 D0  >543            LDA   $D000
4347: C8        >544            INY
4348: 8C 00 D0  >545            STY   $D000
434B: CC 00 D0  >546            CPY   $D000      Read after write (1st)
434E: D0 0C     >547            BNE   :5
4350: EE 00 D0  >548            INC   $D000
4353: C8        >549            INY
4354: CC 00 D0  >550            CPY   $D000      Read after increment (2nd)
4357: D0 03     >551            BNE   :5
4359: E8        >552            INX
435A: 8D 00 D0  >553  :5        STA   $D000
435D: BD 92 43  >554            LDA   CFA,X
4360: A2 00     >555            LDX   #0
4362: F0 0D     >556            BEQ   :4
4364: C9 04     >557  :3        CMP   #4         Apple //c or //gs?
4366: A9 C0     >558            LDA   #$C0
4368: A2 80     >559            LDX   #$80
436A: B0 05     >560            BCS   :4         Yes
436C: 20 42 44  >561            JSR   TEST2E

Unknown label in line: 297 >562
                >562  :4        STA   MEMORY

Unknown label in line: 297 >563
                >563            STX   VID80C
4373: 60        >564            RTS
                >565
4374: EA 2D E6  >566  MACMAT    HEX   EA2DE6E7F9060502
437C: 00        >567  MCODE     HEX   00         Apple 2+
437D: 40 41 42  >568            HEX   404142     Apple //e
4380: 80 81 82  >569            HEX   80818283   Apple //c
4384: C0 C1 C2  >570            HEX   C0C1C2C3C4C5 Apple //gs
438A: 80 80 C0  >571  CFM       HEX   8080C0C0
438E: 00 80 80  >572  CFA       HEX   008080C0
                >573
4392: 05 07 0B  >574  DATA1IDX  DFB   5,7,11,12,17,251
4398: 38 18 01  >575  DATA1VAL  HEX   3818012000D6
                >576  * Routine to detect a mouse card
439E: A2 C7     >577  MOUSEDET LDX   #$C7
43A0: 86 07     >578            STX   AUXPTR+1

Unknown label in line: 297 >579
                >579            STX   MOSL       ;b7 of MOSL set to 1
43A4: 64 06     >581            STZ   AUXPTR

Unknown label in line: 297 >582
                >582            STZ   MOCN

Unknown label in line: 297 >583
```

```
                >583            STZ   MON0
43AA: A2 05     >590  ]LOOP     LDX   #DATA1VAL-DATA1IDX-1
43AC: BC 96 43  >591  ]LOOP1    LDY   DATA1IDX,X
43AF: BD 9C 43  >592            LDA   DATA1VAL,X
43B2: 51 06     >593            EOR   (AUXPTR),Y
43B4: D0 44     >594            BNE   :1
43B6: CA        >595            DEX
43B7: 10 F3     >596            BPL   ]LOOP1
43B9: A5 07     >597            LDA   AUXPTR+1

Unknown label in line: 297 >598
                >598            STA   MOCN
43BD: 29 0F     >599            AND   #$F

Unknown label in line: 297 >600
                >600            STA   MOSL
43C1: 0A        >602            ASL
43C2: 0A        >602            ASL
43C3: 0A        >602            ASL
43C4: 0A        >602            ASL

Unknown label in line: 297 >604
                >604            STA   MON0
43C7: E8        >605            INX                ;X = 0

Unknown label in line: 297 >606
                >606            CPX   MACHINE      Is host an Apple2 or 2+?
43CA: D0 1E     >607            BNE   :2
                >608  * Time to INITMOUSE..
43CC: A0 19     >609            LDY   #$19         Offset to INIT mouse offset
43CE: B1 06     >610            LDA   (AUXPTR),Y
43D0: 85 06     >611            STA   AUXPTR
43D2: A6 07     >612            LDX   AUXPTR+1

Unknown label in line: 297 >613
                >613            LDY   MON0
43D6: 20 03 44  >614            JSR   :0
43D9: 90 0F     >615            BCC   :2

Unknown label in line: 297 >616
                >616            ROR   MOSL         Let set b7 of mouse slot
43DD: A2
Unknown label in line: 297 >617
                >617  :2        LDX   #OM_INI-OM_DEB
43E0: 64 06     >619            STZ   AUXPTR

Unknown label in line: 297 >624
                >624  ]JLOOP    LDY   OM_DEB,X
43E4: B1 06     >625            LDA   (AUXPTR),Y

Unknown label in line: 297 >626
                >626            STA   OM_DEB,X
43E8: CA        >627            DEX
43E9: 10 F7     >628            BPL   ]JLOOP
43EB: 60        >629            RTS
43EC: A6 07     >630  :1        LDX   AUXPTR+1
43EE: E0 C1     >631            CPX   #$C1
```

```
43F0: C6 07    >632           DEC   AUXPTR+1
43F2: B0 B6    >633           BCS   ]LOOP
43F4: 60       >634  :FIN     RTS
43F5: 6C 06 00 >635  :0       JMP   (AUXPTR)
               >636
               >637  * Routine to copy ROM to bank switched RAM
43F8: A0 00    >638  COPYROM  LDY   #0
43FA: A9 F8    >639           LDA   #$F8
43FC: 84 3C    >640           STY   A1L
43FE: 85 3D    >641           STA   A1L+1
4400: 8D 09 C0 >642           STA   $C009      Write into aux ZP
4403: 84 3C    >643           STY   A1L
4405: 85 3D    >644           STA   A1L+1
4407: 8D 08 C0 >645           STA   $C008      Write back into main ZP
440A: 2C 89 C0 >646           BIT   $C089      Write into LC ram
440D: 2C 89 C0 >647           BIT   $C089
4410: B1 3C    >648  ]LOOP    LDA   (A1L),Y
4412: 91 3C    >649           STA   (A1L),Y    within main memory
4414: 8D 09 C0 >650           STA   $C009      Write into aux memory LC bank
4417: 91 3C    >651           STA   (A1L),Y
4419: 8D 08 C0 >652           STA   $C008      Back to writing to main memory
441C: C8       >653           INY
441D: D0 F1    >654           BNE   ]LOOP
441F: E6 3D    >655           INC   A1L+1
4421: A5 3D    >656           LDA   A1L+1
4423: 8D 09 C0 >657           STA   $C009
4426: 85 3D    >658           STA   A1L+1
4428: 8D 08 C0 >659           STA   $C008
442B: D0 E3    >660           BNE   ]LOOP
442D: 2C 81 C0 >661           BIT   $C081
4430: 2C 81 C0 >662           BIT   $C081
4433: 60       >663           RTS
               >664
               >665  * Routine to test //e configuration: 80 col. card?
               >666  * memory expansion?
4434: 08       >667  TEST2E   PHP
4435: 78       >668           SEI
4436: A2 00    >669           LDX   #0
4438: AD 17 C0 >670           LDA   $C017
443B: 30 7D    >671           BMI   :6
443D: E8       >672           INX
443E: AD 1D C0 >673           LDA   $C01D
4441: 48       >674           PHA
4442: AD 18 C0 >675           LDA   $C018
4445: 48       >676           PHA
4446: AD 1C C0 >677           LDA   $C01C
4449: 48       >678           PHA
444A: AD 19 C0 >679  ]LOOP    LDA   $C019
444D: 30 FB    >680           BMI   ]LOOP
444F: 8D 57 C0 >681           STA   $C057
4452: 8D 01 C0 >682           STA   $C001
4455: 8D 55 C0 >683           STA   $C055
4458: AD 00 04 >684           LDA   $400
445B: 48       >685           PHA
445C: AD 00 24 >686           LDA   $2400
445F: 48       >687           PHA
4460: A9 EE    >688           LDA   #$EE
```

```
4462: 8D 00 04 >689           STA   $0400
4465: AD 00 24 >690           LDA   $2400
4468: C9 EE    >691           CMP   #$EE
446A: D0 19    >692           BNE   :2
446C: 0E 00 24 >693           ASL   $2400
446F: AD 00 04 >694           LDA   $0400
4472: CD 00 24 >695           CMP   $2400
4475: F0 29    >696           BEQ   :3
4477: E8       >697  :2       INX
4478: A9 0F    >698           LDA   #$0F
447A: 8D B9 C0 >699           STA   $C0B9
447D: 8D 54 C0 >700           STA   $C054
4480: AD 00 04 >701           LDA   $0400
4483: 8D 00 04 >702           STA   $0400
4486: 8D B8 C0 >703           STA   $C0B8
4489: 8D 55 C0 >704           STA   $C055
448C: AD 00 04 >705           LDA   $0400
448F: 30 0F    >706           BMI   :3
4491: E8       >707           INX
4492: 68       >708  :3       PLA
4493: 8D 00 24 >709           STA   $2400
4496: 68       >710           PLA
4497: 8D 00 04 >711           STA   $0400
449A: 68       >712           PLA
449B: 30 11    >713           BMI   :4
449D: 8D 54 C0 >714           STA   $C054
44A0: 68       >715  :4       PLA
44A1: 30 11    >716           BMI   :5
44A3: 8D 00 C0 >717           STA   $C000
44A6: 68       >718  :5       PLA
44A7: 30 11    >719           BMI   :6
44A9: 8D 56 C0 >720           STA   $C056
               >721  * X=0: No 80 col. card in aux. slot
               >722  * X=1: 80 col. card w/o memory expansion
               >723  * X=2: 80 col. card with at least 64K mem. expansion
               >724  * X=3: Same as above + special video modes (Eve le chat mau
ve)
44AC: BD 8E 43 >725  :6       LDA   CFM,X
44AF: 48       >726           PHA
44B0: BD 92 43 >727           LDA   CFA,X
44B3: AA       >728           TAX
44B4: 68       >729           PLA
44B5: 28       >730           PLP
44B6: 60       >731           RTS
                298           PUT   PEERAUXINSTALL
               >1    STRNG2   EQU   $AD
               >2    FRETOP   EQU   $6F
               >3    HIMEM    EQU   $73
               >4    ALTZP    EQU   $C009
               >5    STDZP    EQU   $C008
               >6    RD80STOR EQU   $C018
               >7    RDLCRAM  EQU   $C012
               >8    RDLCBNK2 EQU   $C011
               >9    GARBAG   EQU   $E484
               >10
               >11   INITBF   STID  CODE1BF;A1L
44B7: A9 3C    >11            LDA   #CODE1BF
```

```
44B9: 85 3C    >11            STA   A1L
44BB: A9 45    >11            LDA   #>CODE1BF
44BD: 85 3D    >11            STA   A1L+1
44BF: A0 00    >12            LDY   #0
44C1: A9 00    >13            LDA   #GZAUXRT
44C3: 85 3E    >13            STA   A2L
44C5: A9 BF    >13            LDA   #>GZAUXRT
44C7: 85 3F    >13            STA   A2L+1
44C9: 8D 05 C0 >14            STA   $C005
44CC: B1 3C    >15   ]LOOP    LDA   (A1L),Y
44CE: 91 3E    >16            STA   (A2L),Y
44D0: C8       >17            INY
44D1: C0 BC    >18            CPY   #CODE2BF-CODE1BF
44D3: D0 F7    >19            BNE   ]LOOP
44D5: 8D 04 C0 >20            STA   $C004
44D8: 08       >21            PHP
44D9: 08       >22            PHP
44DA: 68       >23            PLA
44DB: 78       >24            SEI
44DC: BA       >25            TSX
44DD: 8E 09 C0 >26            STX   ALTZP
44E0: 8E 00 01 >27            STX   $0100
44E3: A2 FF    >28            LDX   #$FF
44E5: 9A       >29            TXS
44E6: 8E 01 01 >30            STX   $0101
44E9: 29 04    >31            AND   #%100
44EB: D0 01    >32            BNE   *+3
44ED: 58       >33            CLI
44EE: A9 F8    >34            LDA   #CODE1LC
44F0: 85 3C    >34            STA   A1L
44F2: A9 45    >34            LDA   #>CODE1LC
44F4: 85 3D    >34            STA   A1L+1
44F6: A9 00    >35            LDA   #$D000
44F8: 85 3E    >35            STA   A2L
44FA: A9 D0    >35            LDA   #>$D000
44FC: 85 3F    >35            STA   A2L+1
44FE: 2C 81 C0 >36            BIT   $C081
4501: 2C 81 C0 >37            BIT   $C081
4504: B2 3C    >39   ]LOOP    LDA   (A1L)
4506: 92 3E    >40            STA   (A2L)
4508: E6 3C    >46            INC   A1L
450A: D0 02    >47            BNE   *+4
450C: E6 3D    >48            INC   A1L+1
450E: A5 3C    >49            LDA   A1L
4510: C9 7B    >50            CMP   #CODE2LC
4512: A5 3D    >51            LDA   A1L+1
4514: E9 46    >52            SBC   #>CODE2LC
4516: B0 16    >53            BCS   :0
4518: E6 3E    >54            INC   A2L
451A: D0 E8    >55            BNE   ]LOOP
451C: E6 3F    >56            INC   A2L+1
451E: 90 E4    >57            BCC   ]LOOP      Always
4520: 78       >58   :0       SEI
4521: BA       >59            TSX
4522: 8E 01 01 >60            STX   $0101
4525: AE 00 01 >61            LDX   $0100
4528: 9A       >62            TXS
```

```
4529: 8E 08 C0 >63            STX   STDZP
452C: 28       >64            PLP
452D: 60       >65   ]RET     RTS
               >66
               >67   CODE1BF  ORG   $BF00
               >68   AXHIMEM  EQU   *
               >69   * Routine de redirection pour la gestion des tableaux en
               >70   * memoire auxiliaire.
               >71   * X:0 init the auxilary memory segment for storing
               >72   *     array elements
               >73   * X:1 check that enough room exists for storing an
               >74   *     array´s elements
               >75   * X:2 actually updates the STREND new array end and
               >76   *     initializes the area.
               >77   * X:3 returns the mem bank free space after a garbage c.
               >78   * X:4 retrieve an array´s element from memory.
               >79   * X:5 stores an array´s element into memory
BF00: BC B8 BF >80   GZAUXRT  LDY   ZAUXOFFT,X offset into Y
BF03: A9 00    >81            LDA   #0
BF05: 2C 12 C0 >82            BIT   RDLCRAM
BF08: 10 09    >83            BPL   *+11
BF0A: 09 0C    >84            ORA   #12
BF0C: 2C 11 C0 >85            BIT   RDLCBNK2
BF0F: 10 02    >86            BPL   *+4
BF11: 49 06    >87            EOR   #6
BF13: 48       >88            PHA
BF14: 08       >89            PHP                 ;Save I bit flag on main stk
BF15: 68       >90            PLA                 ;Restore in b2 of accum.
BF16: BA       >91            TSX
BF17: 78       >92            SEI
BF18: 8D 09 C0 >93            STA   ALTZP
BF1B: 8E 00 01 >94            STX   $0100
BF1E: A2 FF    >95            LDX   #$FF
BF20: 8E 01 01 >96            STX   $0101
BF23: 9A       >97            TXS
BF24: 29 04    >98            AND   #%100         bit I mask
BF26: D0 01    >99            BNE   *+3
BF28: 58       >100           CLI
BF29: AD 18 C0 >101           LDA   RD80STOR
BF2C: 48       >102           PHA
BF2D: 8D 00 C0 >103           STA   $C000         Enable basic access to screens
               >104  * Read/Write enable LC bank 2 in aux. mem. bec. of ALTZP
BF30: 20 A4 BF >105           JSR   G83           Read/Write enable LC bank 2 in
BF33: F4 3D BF >107           PEA   ZAUXRET-1
BF36: 18       >108           CLC
BF37: FB       >109           XCE
BF38: C2 20    >110           REP   $20
               >111           MX    %01
BF3A: 18       >118           CLC
BF3B: 4C 13 D0 >119           JMP   ZAUXRT
               >120
               >121  * Routine de retour general vers le composant principal
               >122  * de Peersoft (en memoire principale)
BF3E: 7A       >124  ZAUXRET  PLY                 ;Restore RD80STOR status
BF3F: 10 03    >128           BPL   *+5           from aux stack..
BF41: 8E 01 C0 >129           STX   $C001         If On, then set it back..
BF44: 08       >130           PHP                 ;Save carry flag
```

```
BF45: 38       >132           SEC             ;Return to emulation mode
BF46: FB       >133           XCE             ; for 65802/816
               >134           MX    %11
BF47: 8A       >135           TXA
BF48: 28       >137           PLP             ;Restore carry flag
BF49: AE 00 01 >138           LDX   $0100     Get back main stack pointer
BF4C: 9A       >139           TXS             ; from $0100 aux stack byte
BF4D: 8E 08 C0 >140           STX   STDZP     Return to Std stack/p0
BF50: AA       >142           TAX
BF51: 68       >144           PLA             ;Restore configuration flag
BF52: 08       >145           PHP             ;Carry back into main stack
BF53: 20 AB BF >146           JSR   G81
BF56: 0A       >147           ASL
BF57: F0 0E    >148           BEQ   :0
BF59: A0 05    >149           LDY   #5
BF5B: BE B2 BF >150  ]LOOP    LDX   IRQTBLE,Y
BF5E: 88       >151           DEY
BF5F: 0A       >152           ASL
BF60: 90 03    >153           BCC   *+5
BF62: 9D 00 C0 >154           STA   $C000,X
BF65: D0 F4    >155           BNE   ]LOOP
BF67: 28       >156  :0       PLP
               >157  * X set to zero upon return according to carry flag
BF68: A2 00    >158           LDX   #0
BF6A: 90 01    >159           BCC   *+3
BF6C: E8       >160           INX
BF6D: 68       >161           PLA             ;Get return address
BF6E: 7A       >163           PLY             ; from main stack
BF6F: 1A       >164           INC
BF70: 8D ED 03 >165           STA   $03ED
BF73: D0 01    >166           BNE   *+3
BF75: C8       >167           INY
BF76: 8C EE 03 >168           STY   $03EE
BF79: 18       >179           CLC
BF7A: B8       >180           CLV
BF7B: 4C 14 C3 >181           JMP   XFER      Retour a l´envoyeur
               >182
BF7E: 08       >183  ZGCPARMS PHP
BF7F: 78       >184           SEI
BF80: 8E 08 C0 >185           STX   STDZP
BF83: A5 AD    >187           LDA   STRNG2
               >188           MX    %01
BF85: 69 07 00 >189           ADC   #$0007    Carry clear upon entry
BF88: 8E 09 C0 >190           STX   ALTZP
BF8B: 90 03    >191           BCC   :0
BF8D: 28       >192           PLP
BF8E: 38       >193           SEC
BF8F: 60       >194  ]RET     RTS
BF90: 28       >207  :0       PLP
BF91: 18       >208           CLC
BF92: 60       >209  ]RET     RTS
               >210
BF93: 8E 08 C0 >211  ZGCP2    STX   STDZP
BF96: 85 AE    >215           STA   STRNG2+1
BF98: 86 AD    >216           STX   STRNG2
BF9A: 8E 09 C0 >218           STX   ALTZP
BF9D: 60       >219           RTS
```

```
                >220
                >222  * Called in emulation mode
BF9E: 20 AB BF  >224  ZNG      JSR   G81
BFA1: 20 84 E4  >225           JSR   GARBAG
BFA4: 2C 83 C0  >226  G83      BIT   $C083
BFA7: 2C 83 C0  >227           BIT   $C083
BFAA: 60        >228           RTS
BFAB: 2C 81 C0  >229  G81      BIT   $C081
BFAE: 2C 81 C0  >230           BIT   $C081
BFB1: 60        >232           RTS
                >233
BFB2: 83 8B 8B  >234  IRQTBLE  HEX   838B8B
BFB5: 05 03 55  >235           HEX   050355
BFB8: 00 15     >236  ZAUXOFFT DFB   ZAUXRT0-ZAUXB,ZAUXRT1-ZAUXB
BFBA: 1D E8     >237           DFB   ZAUXRT2-ZAUXB,ZAUXRT3-ZAUXB
                >241           ERR   */$C000
                >242           ORG
                >243  CODE2BF
                >244  CODE1LC  ORG   $D000
                >245  * Y offset correspondant a X
                >246  * X:0 init the auxilary memory segment for storing
                >247  *     array elements
                >248  * X:1 check that enough room exists for storing an
                >249  *     array´s elements
                >250  * X:2 actually updates the STREND new array end and
                >251  *     initializes the area.
                >252  * X:3 returns the mem bank free space after a garbage c.
                >253  * X:4 retrieve an array´s element from memory.
                >254  * X:5 stores an array´s element into memory
                >255
                >256  * Returns amount of free space in aux memory bank
                >257  * after calling ROM based garbage collection.
D000: E2 20     >259  ZAUXRT3  SEP   $20
D002: 20 9E BF  >260           JSR   ZNG        Call routine within main 48K
D005: C2 20     >261           REP   $20        (mandatory for calling a ROM
                >262           MX    %01
D007: 38        >263           SEC              ; routine)
D008: A5 6F     >264           LDA   FRETOP
D00A: E5 6D     >265           SBC   STREND
D00C: 08        >266           PHP
D00D: 78        >267           SEI
D00E: 20 93 BF  >268           JSR   ZGCP2
D011: 28        >281           PLP
D012: 60        >282  ]RET     RTS
                >283
D013: 8C 17 D0  >284  ZAUXRT   STY   *+4
D016: D0 00     >285           BNE   ZAUXRT0
                >286  * User subroutine is called with Aux mem. stack/p0,
                >287  * 16bits Accu/mem access if 65802/816.
                >288  * Stack pointer set to $FD (a return address ZAUXRET)
                >289  ZAUXB    EQU   *
                >290
                >291  * Do the init
D018: AD 79 D0  >293  ZAUXRT0  LDA   AXARTAB
D01B: 85 69     >294           STA   VARTAB
D01D: 85 6B     >295           STA   ARYTAB
D01F: 85 6D     >296           STA   STREND
```

```
                >297            MX    %01
D021: A9 00 BF  >298            LDA   #AXHIMEM
D024: 85 73     >299            STA   HIMEM
D026: 85 6F     >300            STA   FRETOP
D028: A2 55     >322            LDX   #$55       Pour le Garbage collector...
D02A: 86 52     >323            STX   $52
D02C: 60        >324  ]RET      RTS
                >325
                >326  * Ensure enough room within array segment
D02D: 20 6B D0  >327  ZAUXRT1   JSR   ZCOMRT12
D030: B0 FA     >328            BCS   ]RET
D032: C5 6F     >329            CMP   FRETOP
D034: 60        >334  ]RET      RTS
                >335
D035: A5 6D     >336  ZAUXRT2   LDA   STREND
D037: 85 3C     >337            STA   A1L
D039: 20 6B D0  >342            JSR   ZCOMRT12
D03C: B0 F6     >343            BCS   ]RET
D03E: A0 02     >344            LDY   #2
D040: 85 6D     >346            STA   STREND
D042: 38        >347            SEC
D043: E5 3C     >348            SBC   A1L
D045: 91 3C     >349            STA   (A1L),Y    Offset to next array
                >364  * Temporarily go back to mem/acc. 8bits
D047: E2 20     >365            SEP   $20
                >366            MX    %11
                >368  * # of dimensions
D049: A9 01     >369            LDA   #1
D04B: A0 04     >370            LDY   #4
D04D: 91 3C     >371            STA   (A1L),Y
                >372  * Init segment where elms will be stored
D04F: A2 00     >373            LDX   #0
D051: A4 3C     >374            LDY   A1L
D053: 86 3C     >375            STX   A1L
D055: C4 6D     >376  ]LOOP     CPY   STREND
D057: A5 3D     >377            LDA   A1L+1
D059: E5 6E     >378            SBC   STREND+1
D05B: B0 0A     >379            BCS   *+12
D05D: 8A        >380            TXA
D05E: 91 3C     >381            STA   (A1L),Y
D060: C8        >382            INY
D061: D0 F2     >383            BNE   ]LOOP
D063: E6 3D     >384            INC   A1L+1
D065: 90 EE     >385            BCC   ]LOOP      Always
D067: 18        >386            CLC
                >388  * Go back to 16bits mem/accum.
D068: C2 20     >389            REP   $20
                >390            MX    %01
D06A: 60        >392  ]RET      RTS
                >393
                >466
D06B: 20 7E BF  >467  ZCOMRT12  JSR   ZGCPARMS
D06E: B0 FA     >468            BCS   ]RET
D070: 65 6D     >469            ADC   STREND
                >471            MX    %11
D072: 60        >480  ]RET      RTS
                >481
```

```
                        >482
D073: FF 80 80          >483  TELMS    HEX   FF8080808000
D079: 00 08             >484  AXARTAB  DA    $0800      0
                        >485  AXARYPNT EQU   AXARTAB    2
D07B: 00 00             >486  AXOFFSET DS    2
D07D: 00                >487  ELMSIZ   DS    1          2
D07E: 00 00 00          >488  AXVALUE  DS    5
                        >489  AXARYPT2 EQU   AXVALUE
                        >490  *ZAUXRTF EQU *
                        >491          ORG
                        >492  CODE2LC  EQU   *
                         299          PUT   PEERFGC
                        >1    * Fast garbage collector
                        >2    * Credits: Randy Wiggington
                        >3    STRNG    EQU   $19
                        >4    XXSAV    EQU   $1B
                        >5    PTR2     EQU   $1C
                        >6    DSCLEN   EQU   $8F
                        >7    NUMELS   =     8
                        >8    NUMELS2  =     NUMELS*2
                        >9
466D: A0 00             >10   INITLC   LDY   #0
                        >12           MX    %11
466F: A9 96             >14           LDA   #CODE1GC
4671: 85 3C             >14           STA   A1L
4673: A9 46             >14           LDA   #>CODE1GC
4675: 85 3D             >14           STA   A1L+1
4677: A9 CF             >15           LDA   #CODE1GCF
4679: 85 3E             >15           STA   A2L
467B: A9 47             >15           LDA   #>CODE1GCF
467D: 85 3F             >15           STA   A2L+1
467F: 84 42             >16           STY   A4L
4681: A9 D0             >17           LDA   #>$D000
4683: 85 43             >18           STA   A4L+1
4685: 4C 2C FE          >19           JMP   MOVE
                        >20
                        >21   CODE1GC  ORG   $D000
D000: 18                >23           CLC
D001: FB                >24           XCE
D002: C2 20             >25           REP   $20
D004: A5 73             >26           LDA   HIMEM
D006: 85 6F             >27   FNDVAR   STA   FRETOP
D008: 4C 27 D1          >28           JMP   NZTAB
D00B: 4C EC D0          >29   ]RET     JMP   GRBPAS
D00E: A5 6D             >30   FNDVARX2 LDA   STREND
                        >31           MX    %01
D010: A9 55 00          >32           LDA   #$0055
D013: 85 5E             >33           STA   INDEX
D015: A0 00             >34           LDY   #0
D017: A6 5E             >35   ]LOOP    LDX   INDEX
D019: E4 52             >36           CPX   $52
D01B: F0 05             >37           BEQ   SVARS
D01D: 20 8B D0          >38           JSR   DVAR
D020: F0 F5             >39           BEQ   ]LOOP
D022: A2 07             >40   SVARS    LDX   #7
D024: 86 8F             >41           STX   DSCLEN
D026: A5 69             >42           LDA   VARTAB
```

```
D028: 85 5E    >43            STA   INDEX
D02A: C5 6B    >44   ]LOOP    CMP   ARYTAB
D02C: F0 05    >45            BEQ   ARYVAR
D02E: 20 7C D0 >46            JSR   DVARS
D031: F0 F7    >47            BEQ   ]LOOP
D033: 85 94    >48   ARYVAR   STA   ARYPNT
D035: A0 03    >49            LDY   #3
D037: 84 8F    >50            STY   DSCLEN
D039: 88       >51            DEY                ;Enter loop with Y set to 2
D03A: A5 94    >52   ]LOOP    LDA   ARYPNT
D03C: C5 6D    >53   ]LOOP1   CMP   STREND
D03E: F0 CB    >54            BEQ   ]RET
D040: 85 5E    >55            STA   INDEX
D042: B2 5E    >56            LDA   (INDEX)
               >57            MX    %01
D044: 29 80 80 >58            AND   #$8080
D047: 49 00 80 >59            EOR   #$8000
D04A: 08       >60            PHP                ;Z set iif string
D04B: B1 5E    >61            LDA   (INDEX),Y
D04D: 65 94    >62            ADC   ARYPNT       Carry flag clear
D04F: 85 94    >63            STA   ARYPNT
D051: 28       >64            PLP
D052: D0 E6    >65            BNE   ]LOOP
D054: E2 20    >66            SEP   $20
               >67            MX    %11
D056: A0 04    >68            LDY   #4
D058: B1 5E    >69            LDA   (INDEX),Y
D05A: AA       >70            TAX
D05B: 29 38    >71            AND   #%00111000
D05D: 08       >72            PHP
D05E: 8A       >73            TXA
D05F: 29 07    >74            AND   #7
D061: 1A       >75            INC                ;# of dims in A
D062: A0 00    >76            LDY   #0
D064: 0A       >77            ASL                ;Two bytes per dimension
D065: 28       >78            PLP
               >79   * Either add 5 or 11 according to location of array
               >80   * Main or auxiliary memory
D066: F0 03    >81            BEQ   *+5          Branch iif array in main
D068: 69 0B    >82            ADC   #5+6
D06A: 2C       >83            HEX   2C           Skip next two bytes
D06B: 69 05    >84            ADC   #5
D06D: C2 20    >85            REP   $20
D06F: 65 5E    >86            ADC   INDEX        Add to curr. arr. base addr.
D071: 85 5E    >87            STA   INDEX        and update it
               >88   * End of array?
D073: C5 94    >89   ]LOOP    CMP   ARYPNT
D075: F0 C5    >90            BEQ   ]LOOP1
D077: 20 8B D0 >91            JSR   DVAR
D07A: F0 F7    >92            BEQ   ]LOOP
               >93
               >94   * Garbage collection for simple variables
D07C: B1 5E    >95   DVARS    LDA   (INDEX),Y    Is it a string var
               >96            MX    %01
D07E: 29 80 80 >97            AND   #$8080
D081: 49 00 80 >98            EOR   #$8000
D084: F0 03    >99            BEQ   *+5          Branch iif string var.
```

```
D086: 4C DF D0 >100  GDVARTS  JMP   DVARTS
D089: C8       >101           INY
D08A: C8       >102           INY
D08B: B1 5E    >103  DVAR     LDA   (INDEX),Y
D08D: AA       >104           TAX   Only       low byte is copied
D08E: F0 F6    >105           BEQ   GDVARTS    Skip Zero length strings
D090: 86 2F    >106           STX   LENGTH
D092: C8       >107           INY
D093: B1 5E    >108           LDA   (INDEX),Y  Get value address
D095: 85 19    >109           STA   STRNG
D097: C5 6F    >110           CMP   FRETOP     Is this above where we are?
D099: B0 EB    >111           BCS   GDVARTS    This one´s been collected before
D09B: CD 39 D1 >112           CMP   BTMEL      Compare to lowest value in list
D09E: 90 3F    >113           BCC   DVARTS     No, below lowest, go to next one
D0A0: C5 6D    >114           CMP   STREND
D0A2: 90 E2    >115           BCC   GDVARTS    Inside the program...
D0A4: A2 11    >116           LDX   #NUMELS2+1 Search thru list of elements
D0A6: CA       >117  ]LOOP    DEX
D0A7: CA       >118           DEX
D0A8: DD 38 D1 >119           CMP   BTMEL-1,X
D0AB: 90 F9    >120           BCC   ]LOOP
D0AD: 86 1B    >121           STX   XXSAV
D0AF: A2 03    >122           LDX   #3         Make room in table for entry
D0B1: BD 38 D1 >123  ]LOOP    LDA   BTMEL-1,X
D0B4: 9D 36 D1 >124           STA   BTMEL-3,X  Ripple down
D0B7: BD 48 D1 >125           LDA   LENTHS-1,X
D0BA: 9D 46 D1 >126           STA   LENTHS-3,X
D0BD: BD 58 D1 >127           LDA   VARPT-1,X
D0C0: 9D 56 D1 >128           STA   VARPT-3,X
D0C3: E4 1B    >129           CPX   XXSAV
D0C5: E8       >130           INX
D0C6: E8       >131           INX
D0C7: 90 E8    >132           BCC   ]LOOP
D0C9: A6 1B    >133           LDX   XXSAV
D0CB: A5 19    >134           LDA   STRNG
D0CD: 9D 38 D1 >135           STA   BTMEL-1,X
D0D0: A5 5E    >136           LDA   INDEX
D0D2: 9D 58 D1 >137           STA   VARPT-1,X
D0D5: A5 2F    >138           LDA   LENGTH
D0D7: 9D 49 D1 >139           STA   LENTHS,X
D0DA: A5 8F    >140           LDA   DSCLEN
D0DC: 9D 48 D1 >141           STA   LENTHS-1,X
D0DF: A5 8F    >142  DVARTS   LDA   DSCLEN
D0E1: 29 FF 00 >143           AND   #$00FF
D0E4: 18       >144           CLC
D0E5: 65 5E    >145           ADC   INDEX
D0E7: 85 5E    >146           STA   INDEX
D0E9: A0 00    >147           LDY   #0
D0EB: 60       >148  ]RET     RTS
               >248
               >249  * Have made a complete pass thru the variables
               >250  * Now collect the ones in the list
D0EC: A2 0F    >251  GRBPAS   LDX   #NUMELS2-1
D0EE: BC 49 D1 >253  ]LOOP    LDY   LENTHS,X
D0F1: F0 F8    >254           BEQ   ]RET
D0F3: 64 1C    >255           STZ   PTR2       16bits clear
D0F5: 84 1C    >256           STY   PTR2       8bits setting..
```

```
D0F7: A5 6F    >257           LDA   FRETOP
D0F9: 38       >258           SEC
D0FA: E5 1C    >259           SBC   PTR2
D0FC: 85 6F    >260           STA   FRETOP
D0FE: BD 38 D1 >261           LDA   BTMEL-1,X
D101: 85 1C    >262           STA   PTR2
D103: E2 20    >263           SEP   $20
               >264           MX    %11
D105: 88       >281  ]LOOP1   DEY
D106: C0 FF    >282           CPY   #$FF
D108: F0 06    >283           BEQ   *+8
D10A: B1 1C    >284           LDA   (PTR2),Y
D10C: 91 6F    >285           STA   (FRETOP),Y
D10E: 90 F5    >286           BCC   ]LOOP1     Always
D110: BD 48 D1 >287           LDA   LENTHS-1,X Get size of variable
D113: 29 04    >288           AND   #4
D115: 4A       >289           LSR
D116: A8       >290           TAY
D117: C8       >291           INY
D118: C2 20    >293           REP   $20
               >294           MX    %01
D11A: BD 58 D1 >296           LDA   VARPT-1,X
D11D: 85 1C    >297           STA   PTR2
D11F: A5 6F    >302           LDA   FRETOP
D121: 91 1C    >303           STA   (PTR2),Y
D123: CA       >309           DEX
D124: CA       >310           DEX
D125: 10 C7    >311           BPL   ]LOOP
               >313           MX    %01
D127: A2 0F    >315  NZTAB    LDX   #NUMELS2-1
               >317           MX    %01
D129: A9 00 00 >318           LDA   #$0000
D12C: 9D 48 D1 >319  ]LOOP    STA   LENTHS-1,X
D12F: 9D 38 D1 >320           STA   BTMEL-1,X
D132: CA       >321           DEX
D133: CA       >322           DEX
D134: 10 F6    >323           BPL   ]LOOP
D136: 4C 0E D0 >331           JMP   FNDVARX2
               >413           MX    %11
               >415           DUMMY *
D139: 00 00 00 >416  BTMEL     DS    NUMELS*2
D149: 00 00 00 >417  LENTHS    DS    NUMELS*2
D159: 00 00 00 >418  VARPT     DS    NUMELS*2
               >419           DEND
               >420           ORG
               >421  CODE1GCF EQU   *
                300  * Here is the Peersoft real origine
                305  AROMBA   ORG   $9816-$56-$37-$4C-$B7-$26-$5C-$4AD-$D6-$15DB

                310  FNDVAR2
                311  CGARBAG
                312
                313  * All calls to CHRGET fall into this routine
7AA6: 86 B4     314  DEBUTGET STX   XSAV
7AA8: 84 B5     315           STY   YSAV
                316  * Check return address
7AAA: A3 02     318           LDA   2,S        hi byte
```

```
7AAC: 85 C2     319           STA   OFFSET
7AAE: A3 01     320           LDA   1,S         lo byte
7AB0: A2
Unknown label in line: 327
                327           LDX   #ADAPFTET-ADAPFBET

Unknown label in line: 328
                328  ]LOOP    CMP   ADAPFBET-1,X
7AB5: D0 07     329           BNE   :0

Unknown label in line: 330
                330           LDY   ADAPFTET-1,X
7AB9: C4 C2     331           CPY   OFFSET      Test for a match upon
7ABB: F0 21     332           BEQ   OKP1GET     return address: proceed
7ABD: CA        333  :0       DEX                ;No match: loop till
7ABE: D0 F3     334           BNE   ]LOOP        all values exhaustion
7AC0: A4 B5     335           LDY   YSAV
7AC2: 2C        339           HEX   2C          Skip next two bytes
                341  * No address match: exit with a simulation of CHRGET
7AC3: 86 B4     342  RST100   STX   XSAV
                346  RST101
7AC5: E6 B8     348  LLOOP    INC   TXTPTR
7AC7: D0 03     349           BNE   COMRST
7AC9: E6 B9     350           INC   TXTPTR+1
                355  RST102
                356  RST103
7ACB: B2 B8     357  COMRST   LDA   (TXTPTR)
7ACD: C9 20     359           CMP   #$20
7ACF: F0 F5     360           BEQ   LLOOP
7AD1: A6 B4     361           LDX   XSAV
7AD3: C9 3A     362  COMRSTC  CMP   #':'
7AD5: B0 06     363           BCS   :0
7AD7: E9 2F     364           SBC   #$30-1      Because of carry clear
7AD9: 38        365           SEC
7ADA: E9 D0     366           SBC   #$D0
7ADC: 60        367  :0       RTS
                372
                373  OKP1GET
                374  * Tricky way to replace the two bytes at the top of stack
                375  * Instead of doing PLA PLA followed by PHA PHA...

Unknown label in line: 377
                377           LDA   ADPFB-1,X
7ADF: 83 01     378           STA   1,S

Unknown label in line: 379
                379           LDA   ADPFT-1,X
7AE3: 83 02     380           STA   2,S
7AE5: D0 DF     390           BNE   RST101      Always
7AE7: 4C 90 7E  391  GNPTRGET JMP   NPTRGET
7AEA: 86 B4     392  DEBUTGOT STX   XSAV
7AEC: BA        393           TSX
7AED: A3 01     395           LDA   1,S
7AEF: C9 EE     399           CMP   #VPTRGET-1
7AF1: D0 D9     400           BNE   RST103
7AF3: A3 02     402           LDA   2,S
7AF5: 49 DF     406           EOR   #>VPTRGET-1 A=0 upon matching address
```

```
7AF7: D0 D3     407           BNE   RST103
7AF9: A3 04     409           LDA   4,S
7AFB: BA        410           TSX
7AFC: E8        411           INX
7AFD: E8        412           INX
7AFE: C9 DA     418           CMP   #>VLET+2
7B00: D0 06     419           BNE   :44
7B02: E8        420           INX
7B03: E8        421           INX             ;Carry set at this time
7B04: 24        422           HEX   24        Skip next byte
7B05: 18        423  :44      CLC
7B06: 9A        424           TXS
7B07: A2 00     425           LDX   #0
7B09: 90 DF     426           BCC   GNPTRGET
                427  * The following routine handles the Applesoft
                428  * variable setting
                429  * (LET is the optional keyword)
7B0B: 20 90 7E  430  RLET     JSR   NPTRGET
7B0E: 85 85     431           STA   FORPNT
7B10: 84 86     432           STY   FORPNT+1
7B12: A6 BF     433           LDX   AUXBANK
7B14: F0 1F     434           BEQ   RLET1
7B16: D4 9B     436           PEI   LOWTR
7B18: D4 AD     437           PEI   STRNG2
7B1A: DA        448           PHX
7B1B: 20 35 7B  449           JSR   RLET1
7B1E: 68        450           PLA
7B1F: 85 BF     451           STA   AUXBANK
7B21: 68        452           PLA
7B22: 85 AD     453           STA   STRNG2
7B24: 68        454           PLA
7B25: 85 AE     455           STA   STRNG2+1
7B27: 68        456           PLA
7B28: 85 9B     457           STA   LOWTR
7B2A: 68        458           PLA
7B2B: 85 9C     459           STA   LOWTR+1
7B2D: A2 05     460           LDX   #5
7B2F: 4C 2F 82  461           JMP   ZRTAUX
                462
7B32: B2 B8     467  RLET1    LDA   (TXTPTR)
7B34: A2 03     469           LDX   #3        New syntax scheme?

Unknown label in line: 470
                470  ]LOOP    CMP   TOKENS,X
7B38: F0 2A     471           BEQ   :0        yes so handle it
7B3A: CA        472           DEX
7B3B: 10 F9     473           BPL   ]LOOP
7B3D: A9 D0     474           LDA   #TOKEQUAL
7B3F: 20 DA 82  475           JSR   NSYNCHR2  Y vaut deja zero si 6502
7B42: A5 12     476           LDA   INTTYP
7B44: 10 18     477           BPL   :11
7B46: 48        478           PHA
7B47: 20 8C 89  479           JSR   NFRMNUM
7B4A: 20 D6 7C  480  ]LOOP    JSR   NROUT
7B4D: 68        481           PLA
7B4E: C9 81     482           CMP   #$81      Byte subtype?
7B50: D0 0F     483           BNE   :12
```

```
7B52: 20 72 7E  484           JSR   CONV1628
7B55: A5 A1     485           LDA   FAC+4
7B57: 92 85     487           STA   (FORPNT)
7B59: 60        492           RTS
7B5A: 4C 52 DA  493  :11      JMP   VLET+12
7B5D: 4C 6B DA  494  :12      JMP   $DA6B
                495
                496  * Save selected operation on stack (+,-,*,/)
                497  :0       MPHX
7B60: DA        497           PHX
7B61: 20 C6 7A  498           JSR   RST101     Bump next character
                499  * Ensure that next char is ´=´ symbol token
7B64: A9 D0     500           LDA   #TOKEQUAL
7B66: 20 DA 82  501           JSR   NSYNCHR2   no need to reset Y to 0
                502  * Save variable type on stack
7B69: A5 12     503           LDA   INTTYP     $80 iif integer variable
7B6B: 48        504           PHA
7B6C: A5 11     505           LDA   VALTYP     $FF iif string
7B6E: 48        506           PHA
7B6F: 20 7B DD  507           JSR   FRMEVL
7B72: 68        508           PLA
7B73: 2A        509           ROL              ;Carry set iif var. type string
7B74: 20 6D DD  510           JSR   $DD6D      Check FRMEVL result type accordin
g to C
7B77: 68        511           PLA              ;Get INTTYP off stack
7B78: B0 23     512           BCS   HNDLESTR   String variable and expression
                513  * From then on: we´ll handle numeric var. and expr.
7B7A: 30 4C     514           BMI   HNDLEINT
7B7C: A4 86     515  HNDLEREA LDY   FORPNT+1
7B7E: 68        516           PLA
7B7F: 0A        518           ASL
7B80: AA        520           TAX
7B81: F4 26 EB  522           PEA   $EB27-1
7B84: A5 85     530           LDA   FORPNT

Unknown label in line: 531
                531           JMP   (FPROUTS,X)
                540
7B88: 4C 27 EB  541  ]LOOP1   JMP   $EB27      SETFOR
7B8B: A5 12     542  NLET2    LDA   INTTYP
7B8D: 10 F9     543           BPL   ]LOOP1
7B8F: 48        544           PHA
7B90: 30 B8     545           BMI   ]LOOP      Always
                546
                547  * Includes module for handling integ. arithmetic
                548  * and <op>= instructions
                549           PUT   PEERINTEGARITH
               >1    * Module handling all integer arithmetic
               >2    * within Peersoft and all op= instructions
               >3    FCOMP    EQU   $EBB2
               >4
7B92: 4C 76 DD >5    ]ERR     JMP   GOTMIERR
7B95: 4C B2 E5 >6    ]ERR1    JMP   GOSTLERR
               >7
               >8    * Handle += instruction for string variables
7B98: 68       >9    HNDLESTR PLA              ;Get OP kind off stack
7B99: D0 F7    >10             BNE   ]ERR       ;Only ADD operation allowed
```

```
7B9B: B2 A0    >18            LDA   ($A0)
7B9D: F0 65    >19            BEQ   RET1
7B9F: 18       >20            CLC
7BA0: 72 85    >21            ADC   (FORPNT)
7BA2: B0 F1    >23            BCS   ]ERR1
7BA4: 20 DD E3 >24            JSR   STRSPA
7BA7: A5 85    >25            LDA   FORPNT
7BA9: A4 86    >26            LDY   FORPNT+1
7BAB: 20 C1 7B >27            JSR   NMOVINS
7BAE: A0 02    >28            LDY   #2
7BB0: B9 9D 00 >29   ]LOOP    LDA   DSCTMP,Y
7BB3: 91 85    >30            STA   (FORPNT),Y
7BB5: 88       >31            DEY
7BB6: 10 F8    >32            BPL   ]LOOP
7BB8: A5 A0    >33            LDA   $A0
7BBA: A4 A1    >34            LDY   $A1
7BBC: 85 AB    >35   NMOVINS  STA   STRING1
7BBE: 84 AC    >36            STY   STRING1+1
7BC0: 4C D4 E5 >37            JMP   MOVINS
               >38
7BC3: 29 07    >39   HNDLEINT AND   #7         Integer subtype in A reg.
7BC5: C9 02    >40            CMP   #2         Correct if 16bits integer
7BC7: D0 07    >41            BNE   :0
7BC9: A9 00    >42            LDA   #0
               >43   * On enclenche NROUT que si 8 ou 16bits
7BCB: C9 02    >44   :0       CMP   #2
7BCD: B0 12    >45            BCS   :1
7BCF: 48       >46            PHA
7BD0: 20 D6 7C >47            JSR   NROUT
7BD3: 68       >48            PLA
7BD4: F0 0C    >49            BEQ   :2
               >50   * Ensure correct value for 8bits integer
7BD6: AA       >51            TAX
7BD7: 20 72 7E >52            JSR   CONV1628
7BDA: 8A       >53            TXA
7BDB: 24       >55            HEX   24         Skip next byte
7BDC: 3A       >56   :1       DEC
7BDD: 0A       >61   :2       ASL
7BDE: 0A       >62            ASL
7BDF: 85 B4    >63            STA   XSAV
7BE1: 68       >64            PLA              ;Retrieve ope. index in A reg.
7BE2: 05 B4    >65            ORA   XSAV

Unknown label in line: 549 >66
               >66            BIT   WMODE
7BE6: 10 02    >67            BPL   *+4
7BE8: 09 08    >68            ORA   #8
7BEA: AA       >69            TAX              ;Global operation offset into X

Unknown label in line: 549 >70
               >70   HNDLEIY  LDA   OFFSTT,X
7BED: 48       >71            PHA

Unknown label in line: 549 >72
               >72            LDA   OFFSTB,X
7BF0: 48       >73            PHA
7BF1: A0 01    >74            LDY   #1
```

```
7BF3: 8A       >75            TXA
7BF4: 29 04    >76            AND   #4
7BF6: F0 01    >77            BEQ   *+3        Branch iif 16bits int operation
7BF8: 88       >78            DEY
7BF9: 18       >79            CLC
7BFA: B1 85    >80            LDA   (FORPNT),Y
7BFC: 60       >84   RET1     RTS
               >85
7BFD: 65 A1    >86   HNDLUIAD ADC   $A1
7BFF: AA       >87            TAX              ;Low byte in X reg.
7C00: B2 85    >89            LDA   (FORPNT)
7C02: 65 A0    >93            ADC   $A0
7C04: 90 55    >94            BCC   HNDLEIC
7C06: 4C D5 E8 >95   ]ERR     JMP   GOOVFERR
7C09: 38       >96   HNDLUIMI SEC
7C0A: E5 A1    >97            SBC   $A1
7C0C: AA       >98            TAX              ;Low byte in X reg.
7C0D: B2 85    >100           LDA   (FORPNT)
7C0F: E5 A0    >104           SBC   $A0
7C11: 90 F3    >105           BCC   ]ERR
7C13: B0 46    >106           BCS   HNDLEIC
7C15: 65 A1    >107  HNDLSIAD ADC   $A1        ADD operation
7C17: AA       >108           TAX
7C18: B2 85    >112           LDA   (FORPNT)
7C1A: 65 A0    >114           ADC   $A0
7C1C: 70 E8    >115           BVS   ]ERR
7C1E: 50 3B    >116           BVC   HNDLEIC
7C20: 38       >117  HNDLSIMI SEC
7C21: E5 A1    >118           SBC   $A1
7C23: AA       >119           TAX
7C24: B2 85    >123           LDA   (FORPNT)
7C26: E5 A0    >125           SBC   $A0
7C28: 70 DC    >126           BVS   ]ERR
7C2A: 50 2F    >127           BVC   HNDLEIC
7C2C: 38       >128  HNDLUIDV SEC
7C2D: 20 B8 7C >129  HNDLUIMU JSR   LBS49
7C30: 90 0D    >130           BCC   :0
7C32: 20 1D 7E >131           JSR   USDIV
7C35: 80 03    >133           BRA   *+5
7C37: 20 CB 7D >137  :0       JSR   USMUL
7C3A: D0 CA    >138           BNE   ]ERR
7C3C: F0 18    >139           BEQ   HNDLEIX
7C3E: 38       >140  HNDLSIDV SEC
7C3F: 20 B8 7C >141  HNDLSIMU JSR   LBS49
7C42: B0 0D    >142           BCS   :0
7C44: 20 AD 7D >143           JSR   SMUL
7C47: 80 03    >145           BRA   *+5
7C49: 20 F2 7D >149  :0       JSR   SDIV
7C4C: 70 B8    >150           BVS   ]ERR
7C4E: C8       >152  HNDLEIX  INY
7C4F: A6 C2    >153           LDX   MPLIER
7C51: A5 C3    >157           LDA   MPLIER+1
7C53: 92 85    >159  HNDLEIC  STA   (FORPNT)
7C55: 8A       >163           TXA              ;Low byte from result
7C56: 91 85    >167           STA   (FORPNT),Y
7C58: A9 80    >168  SETITS   LDA   #$80
7C5A: 85 C7    >169           STA   INTTYPSV
```

```
7C5C: 60       >170           RTS
               >171
7C5D: 65 A1    >172  HNDLUBAD ADC   $A1
7C5F: 90 54    >173           BCC   HNDLEBC
7C61: 4C D5 E8 >174  ]ERR     JMP   GOOVFERR
7C64: 65 A1    >175  HNDLSBAD ADC   $A1
7C66: 70 F9    >176           BVS   ]ERR
               >177  ]ERRS    EQU   *-2
7C68: 50 4B    >178           BVC   HNDLEBC
7C6A: 38       >179  HNDLUBMI SEC
7C6B: E5 A1    >180           SBC   $A1
7C6D: 90 F2    >181           BCC   ]ERR
7C6F: B0 44    >182           BCS   HNDLEBC
7C71: 38       >183  HNDLSBMI SEC
7C72: E5 A1    >184           SBC   $A1
7C74: 70 F0    >185           BVS   ]ERRS
7C76: 50 3D    >186           BVC   HNDLEBC
7C78: 38       >187  HNDLUBMU SEC
7C79: 85 C2    >188  HNDLSBMU STA   MPLIER
7C7B: A5 A1    >189           LDA   $A1
7C7D: 85 C0    >190           STA   MCAND
7C7F: 90 11    >191           BCC   :0
7C81: 20 30 7D >192           JSR   USMUL8
7C84: D0 DB    >193           BNE   ]ERR
7C86: A5 C2    >194           LDA   MPLIER
7C88: 70 2B    >195           BVS   HNDLEBC    Always (see USMUL8 routine)
7C8A: 20 13 7D >196  :0       JSR   SMUL8
7C8D: 70 D7    >197           BVS   ]ERRS
7C8F: A5 C2    >198           LDA   MPLIER
7C91: 50 22    >199           BVC   HNDLEBC    Always
               >200
7C93: 4C E1 EA >201  ]ERR     JMP   GODVZERR
7C96: 38       >202  HNDLUBDV SEC
7C97: 85 C2    >203  HNDLSBDV STA   DIVEND
7C99: A5 A1    >204           LDA   $A1
7C9B: F0 F6    >205           BEQ   ]ERR
7C9D: 85 C0    >206           STA   DIVSOR
7C9F: 90 0D    >207           BCC   :0
7CA1: 20 78 7D >208           JSR   USDIV8
7CA4: 70 0F    >209           BVS   HNDLEBC    Always (see USDIV8 routine)
7CA6: 20 4B 7D >210  :0       JSR   SDIV8
7CA9: 70 BB    >211           BVS   ]ERRS
7CAB: A5 C2    >212           LDA   DIVEND
7CAD: 92 85    >214  HNDLEBC  STA   (FORPNT)
7CAF: 60       >219  ]RET     RTS
               >220
7CB0: 08       >221  LBS49    PHP
7CB1: 85 C2    >222           STA   MPLIER
7CB3: B2 85    >224           LDA   (FORPNT)
7CB5: 85 C3    >228           STA   MPLIER+1
7CB7: A5 A0    >229           LDA   $A0
7CB9: 85 C1    >230           STA   MCAND+1
7CBB: A5 A1    >231           LDA   $A1
7CBD: 85 C0    >232           STA   MCAND
7CBF: 28       >233           PLP
7CC0: 60       >234           RTS
               >235
```

```
7CC1: 4C 99 E1 >236  ]ERR     JMP   $E199
7CC4: 20 F2 EB >238  ]LOOP    JSR   QINT
7CC7: 18       >239           CLC
7CC8: 60       >240           RTS
               >242  * LBS03 is called with carry flag as input parm
               >243  * Carry set: for catering with negative STEP values
               >244  * while unsigned arithmetic is active.
7CC9: 08       >245  LBS03    PHP
7CCA: 20 8C 89 >246           JSR   NFRMNUM
7CCD: 24       >247           HEX   24
7CCE: 08       >248  NROUT    PHP
7CCF: 20 72 EB >249           JSR   $EB72      Arrondit FAC
7CD2: 28       >250           PLP
7CD3: A5 9D    >251  NEWAYINT LDA   FAC

Unknown label in line: 549 >252
               >252           BIT   WMODE
7CD7: 30 13    >253           BMI   :1
7CD9: C9 90    >254           CMP   #$90
7CDB: 90 E7    >256           BCC   ]LOOP
7CDD: 20 BA 92 >260           JSR   GN32768
7CE0: 4C 16 E1 >261           JMP   $E116
               >262  * Unsigned mode
7CE3: 24 A2    >263  :1       BIT   FACSIGN
7CE5: B0 20    >264           BCS   :3
7CE7: 30 D8    >265           BMI   ]ERR
7CE9: C9 91    >266  :2       CMP   #$91
7CEB: 90 D7    >268           BCC   ]LOOP
7CED: 20 BF 92 >272           JSR   GP32768
7CF0: 20 B2 EB >273           JSR   FCOMP
7CF3: A8       >274           TAY
7CF4: 30 CE    >276           BMI   ]LOOP      A = -1 so FAC < 32768
7CF6: 20 C4 92 >280           JSR   GN65536
7CF9: 20 BE E7 >281           JSR   FADD
7CFC: 80 C6    >283           BRA   ]LOOP
7CFE: 10 F2    >289  :3       BPL   :2
7D00: 20 D0 EE >290           JSR   NEGOP
7D03: A5 9D    >291           LDA   FAC
7D05: 20 F2 7C >292           JSR   :2
7D08: 38       >293           SEC
7D09: 60       >294           RTS
               >295
               >296  * Signed 8bits multiplication: result in 8bits
               >297  * with possible overflow exception
               >298  * MCAND and MPLIER set upon entry
               >299  * Result in MPLIER
               >300  * Credits: Randy Hyde
7D0A: A5 C0    >301  SMUL8    LDA   MCAND
7D0C: 45 C2    >302           EOR   MPLIER
7D0E: 48       >303           PHA              ;Bit N set if signs differ
7D0F: 20 93 7D >304           JSR   ZPRT8
7D12: 20 30 7D >305           JSR   USMUL8
7D15: FA       >306           PLX
7D16: 98       >307           TYA
7D17: D0 16    >308           BNE   :0
7D19: A5 C2    >309           LDA   MPLIER
7D1B: 30 12    >310           BMI   :0
```

```
7D1D: 8A       >311           TXA
7D1E: 10 0E    >312           BPL   :1
7D20: A2 C2    >313           LDX   #MPLIER
7D22: 20 A2 7D >314           JSR   NEG8
7D25: B8       >315  :1       CLV
7D26: 60       >316  :0       RTS
               >317
7D27: A0 08    >318  USMUL8   LDY   #8
7D29: A5 C2    >319  ]LOOP    LDA   MPLIER     Get lsb of MPLIER
7D2B: 4A       >320           LSR              ; into C
7D2C: 90 10    >321           BCC   :4
7D2E: 18       >322           CLC
7D2F: A5 BE    >323           LDA   PARTIAL
7D31: 65 C0    >324           ADC   MCAND
7D33: 85 BE    >325           STA   PARTIAL
               >326  * Shift result into MPLIER
7D35: 66 BE    >327  :4       ROR   PARTIAL
7D37: 66 C2    >328           ROR   MPLIER
7D39: 88       >329           DEY              ;All MPLIER 8 bits
7D3A: D0 ED    >330           BNE   ]LOOP       have been processed?
7D3C: 2C AF 7C >331           BIT   ]RET       Bit V set..
7D3F: A4 BE    >332           LDY   PARTIAL
7D41: 60       >333  ]RET     RTS
               >334
               >335  * Signed 8bits integer divide routine
               >336  * with possible overflow and divide by zero exceptions
               >337  * DIVEND and DIVSOR set upon entry
               >338  * Result in DIVEND
               >339  * Credits: Randy Hyde
7D42: A5 C0    >340  SDIV8    LDA   DIVSOR
7D44: 49 80    >341           EOR   #$80
7D46: D0 16    >342           BNE   :1
               >343  * On traite le cas ou le diviseur est -128
               >344  * Dans ce cas la si DIVEND vaut aussi -128, alors
               >345  * retourne 1 sinon 0
7D48: A8       >346           TAY
7D49: AA       >347           TAX              ;X forced to zero
7D4A: A5 C2    >348           LDA   DIVEND
7D4C: C9 80    >349           CMP   #$80
7D4E: D0 01    >350           BNE   *+3
7D50: E8       >351           INX
7D51: 86 C2    >352           STX   DIVEND
7D53: D0 EC    >353           BNE   ]RET
7D55: A5 C0    >354  :1       LDA   DIVSOR
7D57: 45 C2    >355  :2       EOR   DIVEND
7D59: 48       >356           PHA              ;Sign bit on stack
7D5A: 20 93 7D >357           JSR   ZPRT8      ;Absolute value for operands
7D5D: 20 78 7D >358           JSR   USDIV8
7D60: 1A       >362           INC
7D61: F0 13    >364           BEQ   :3         Keep V set and exit
7D63: 68       >365           PLA              ;Get back sign
7D64: 10 05    >366           BPL   *+7        No need to get result opposite
7D66: A2 C2    >367           LDX   #DIVEND
7D68: 20 A2 7D >368           JSR   NEG8
               >369  * Exit with V clear
7D6B: B8       >370           CLV
7D6C: 60       >371           RTS
```

```
7D6D: 68       >372  :3       PLA
7D6E: 60       >373  ]RET     RTS
               >374
7D6F: A0 08    >375  USDIV8   LDY   #8
7D71: 06 C2    >376  ]LOOP    ASL   DIVEND
7D73: 26 BE    >377           ROL   PARTIAL
7D75: 38       >378           SEC
7D76: A5 BE    >379           LDA   PARTIAL
7D78: E5 C0    >380           SBC   DIVSOR
7D7A: AA       >381           TAX
7D7B: 90 0D    >382           BCC   :3
7D7D: 86 BE    >383           STX   PARTIAL
7D7F: E6 C2    >384           INC   DIVEND
7D81: 88       >385  :3       DEY
7D82: D0 ED    >386           BNE   ]LOOP
7D84: 2C 6E 7D >387           BIT   ]RET       V set by default
7D87: A5 C2    >388           LDA   DIVEND
7D89: 60       >389           RTS
               >390
7D8A: A0 00    >391  ZPRT8    LDY   #0
7D8C: 84 BE    >392           STY   PARTIAL
7D8E: A2 C0    >393           LDX   #MCAND
7D90: 20 9E 7D >394           JSR   ABSOL8
7D93: A2 C2    >395           LDX   #MPLIER
7D95: B5 00    >396  ABSOL8   LDA   0,X
7D97: 10 D5    >397           BPL   ]RET
7D99: 98       >398  NEG8     TYA
7D9A: 38       >399           SEC
7D9B: F5 00    >400           SBC   0,X
7D9D: 95 00    >401           STA   0,X
7D9F: 60       >402  ]RET     RTS
               >403
               >404  * Signed 16bits multiplication: result in 16bits
               >405  * with possible overflow exception
               >406  * MCAND and MPLIER set upon entry
               >407  * Result in MPLIER
               >408  * Credits: Randy Hyde
7DA0: 2C 9F 7D >409  ]LOOP    BIT   ]RET
7DA3: 60       >410           RTS
7DA4: A5 C1    >411  SMUL     LDA   MCAND+1
7DA6: 45 C3    >412           EOR   MPLIER+1
7DA8: 48       >413           PHA              ;BitN set if signs differ
7DA9: 20 52 7E >414           JSR   ZEROPRT    Get absolute values of operands
7DAC: 20 CB 7D >415           JSR   USMUL
7DAF: A8       >416           TAY
7DB0: FA       >417           PLX
7DB1: 98       >418           TYA
7DB2: D0 EC    >419           BNE   ]LOOP
7DB4: A5 C3    >420           LDA   MPLIER+1
7DB6: 30 E8    >421           BMI   ]LOOP
7DB8: 8A       >422           TXA
7DB9: 10 0E    >423           BPL   :8
7DBB: A2 C2    >424           LDX   #MPLIER
7DBD: 20 63 7E >425           JSR   NEGATE
7DC0: B8       >426  :8       CLV              ;reset bit V to zero
7DC1: 60       >427  ]RET     RTS
               >428
```

```
7DC2: A0 10    >429  USMUL    LDY   #16
7DC4: A5 C2    >430  ]LOOP    LDA   MPLIER     Get lsb of MPLIER
7DC6: 4A       >431          LSR               ; into C
7DC7: 90 16    >432          BCC   :4
7DC9: 18       >433          CLC
7DCA: A5 BE    >434          LDA   PARTIAL
7DCC: 65 C0    >435          ADC   MCAND
7DCE: 85 BE    >436          STA   PARTIAL
7DD0: A5 BF    >437          LDA   PARTIAL+1
7DD2: 65 C1    >438          ADC   MCAND+1
7DD4: 85 BF    >439          STA   PARTIAL+1
               >440  * Shift result into MPLIER
7DD6: 66 BF    >441  :4       ROR   PARTIAL+1
7DD8: 66 BE    >442          ROR   PARTIAL
7DDA: 66 C3    >443          ROR   MPLIER+1
7DDC: 66 C2    >444          ROR   MPLIER
7DDE: 88       >445          DEY               ;All MPLIER 16 bits
7DDF: D0 E3    >446          BNE   ]LOOP        have been processed?
7DE1: A5 BE    >447          LDA   PARTIAL
7DE3: 05 BF    >448          ORA   PARTIAL+1
7DE5: 60       >449  ]RET     RTS
               >450
7DE6: 4C E1 EA >451  DVZERROR JMP   GODVZERR
               >452  * Signed 16bits integer divide routine
7DE9: A5 C1    >453  SDIV     LDA   DIVSOR+1
7DEB: 05 C0    >454          ORA   DIVSOR
7DED: F0 00    >455          BEQ   DVZERROR
7DEF: A5 C1    >456          LDA   DIVSOR+1
7DF1: C9 80    >457          CMP   #>$8000
7DF3: D0 22    >458          BNE   :2
7DF5: A5 C0    >459          LDA   DIVSOR
7DF7: D0 1C    >460          BNE   :1
               >461  * On traite le cas ou le diviseur est -32768
               >462  * Dans ce cas la si DIVEND vaut aussi -32768, alors
               >463  * retourne 1 sinon 0
7DF9: A8       >464          TAY
7DFA: AA       >465          TAX               ;X forced to zero
7DFB: C5 C2    >466          CMP   DIVEND
7DFD: D0 10    >467          BNE   :0
7DFF: A5 C3    >468          LDA   DIVEND+1
7E01: C9 80    >469          CMP   #>$8000
7E03: D0 0A    >470          BNE   :0
7E05: E8       >471          INX
7E06: 86 C2    >472  :0       STX   DIVEND
7E08: 84 C3    >473          STY   DIVEND+1
7E0A: D0 42    >474          BNE   NRET        Always
7E0C: A5 C1    >475  :1       LDA   DIVSOR+1
7E0E: 45 C3    >476  :2       EOR   DIVEND+1
7E10: 48       >477          PHA               ;Sign bit on stack
7E11: 20 52 7E >478          JSR   ZEROPRT     ;Absolute value for operands
7E14: A0 10    >479  USDIV    LDY   #16
7E16: 06 C2    >480  ]LOOP    ASL   DIVEND
7E18: 26 C3    >481          ROL   DIVEND+1
7E1A: 26 BE    >482          ROL   PARTIAL
7E1C: 26 BF    >483          ROL   PARTIAL+1
7E1E: 38       >484          SEC
7E1F: A5 BE    >485          LDA   PARTIAL
```

```
7E21: E5 C0    >486           SBC   DIVSOR
7E23: AA       >487           TAX
7E24: A5 BF    >488           LDA   PARTIAL+1
7E26: E5 C1    >489           SBC   DIVSOR+1
7E28: 90 0F    >490           BCC   :3
7E2A: 86 BE    >491           STX   PARTIAL
7E2C: 85 BF    >492           STA   PARTIAL+1
7E2E: E6 C2    >493           INC   DIVEND
7E30: 88       >494  :3       DEY
7E31: D0 E3    >495           BNE   ]LOOP
7E33: 2C 51 7E >496           BIT   ARET+1     V set by default
7E36: A5 C2    >497           LDA   DIVEND
7E38: 25 C3    >498           AND   DIVEND+1
7E3A: 1A       >502           INC
7E3B: F0 13    >504           BEQ   ARET       Keep V set and exit
7E3D: 68       >505           PLA              ;Get back sign
7E3E: 10 0E    >506           BPL   NRET       No need to get result opposite
7E40: A2 C2    >507           LDX   #DIVEND
7E42: 20 63 7E >508           JSR   NEGATE
               >509  * Exit with V clear
7E45: B8       >510  NRET     CLV
7E46: 70       >511           HEX   70         Skip next byte
7E47: 68       >512  ARET     PLA
7E48: 60       >513  ]RET     RTS
               >514
               >515  * Zero partial and fall into ABSOPND
7E49: A0 00    >516  ZEROPRT  LDY   #0
7E4B: 84 BE    >517           STY   PARTIAL
7E4D: 84 BF    >518           STY   PARTIAL+1
7E4F: A2 C0    >519           LDX   #MCAND
7E51: 20 5F 7E >520           JSR   ABSOLUTE
7E54: A2 C2    >521           LDX   #MPLIER    ;Fall into ABSOLUTE
               >522  * Compute absolute value of integer pointed to by X
               >523  * in ZP
7E56: B5 01    >524  ABSOLUTE LDA   1,X
7E58: 10 EE    >525           BPL   ]RET       No need
7E5A: 38       >526  NEGATE   SEC
7E5B: 98       >527           TYA              ;Y set to 0 upon entry
7E5C: F5 00    >528           SBC   0,X
7E5E: 95 00    >529           STA   0,X
7E60: 98       >530           TYA
7E61: F5 01    >531           SBC   1,X
7E63: 95 01    >532           STA   1,X
7E65: 60       >533  ]RET     RTS
               >534
               >535  * Conversion from 16bits to 8bits with provision for
               >536  * ILLEGAL QUANTITY..
7E66: 4C 99 E1 >537  ]ERR     JMP   GOIQERR
7E69: A5 A0    >538  CONV1628 LDA   FAC+3      High byte

Unknown label in line: 549 >539
               >539           BIT   WMODE
7E6D: 30 15    >540           BMI   :0
7E6F: A8       >541           TAY
7E70: C8       >542           INY
7E71: C0 02    >543           CPY   #2         Must be either -1 or 0
7E73: B0 F1    >544           BCS   ]ERR        in unsigned mode
```

```
7E75: 45 A1    >545           EOR   FAC+4      b7 of low byte should be
7E77: 30 ED    >546           BMI   ]ERR       set accordingly.
7E79: 60       >547           RTS
7E7A: D0 EA    >548   :0      BNE   ]ERR       Must be zero if unsigned mode
7E7C: 60       >549           RTS
7E7D: 4C 99 E1 >550           JMP   GOIQERR
                550   * New processing for variable lookup
                551           PUT   PEERNPTRGET
               >1     MKNV    EQU   $E09C      Make new variable (ROM routine)
               >2     SETVYA  EQU   $E0DE      Set LOWTR and Y,A if var. found
               >3
7E80: A9 40    >4     NGETARPT LDA  #$40       $40: only look for arrays
7E82: 85 14    >5             STA   SUBFLG
               >6     * This routine is the new PTRGET routine from PEERSOFT
               >7     NPTRGTX
7E84: 64 10    >12            STZ   DIMFLG
               >14    NPTRGET
               >15    * Upon exit from the above routine, the X reg will
               >16    * contain the value X had upon call to CHRGOT (here zero)
7E86: 20 CC 7A >17            JSR   COMRST
               >18    * First variable name character must be alphabetic
7E89: 20 D4 82 >19            JSR   MISLETC
               >20
7E8C: 64 11    >28    NPTRGET1 STZ  VALTYP
7E8E: 64 12    >29            STZ   INTTYP
7E90: 64 82    >30            STZ   VARNAM+1   Default zero for 2nd name char.
7E92: 64 BF    >31            STZ   AUXBANK
7E94: 85 81    >33            STA   VARNAM
7E96: 20 C4 7A >34            JSR   RST100
7E99: 90 0F    >35            BCC   GTLT       Branch if numeric digit
7E9B: 20 7D E0 >36            JSR   ISLETC
7E9E: 90 24    >37            BCC   EXPLIC?    Branch if not alpha character
7EA0: AA       >38    GTLT    TAX              ;2nd character in X
7EA1: 86 82    >39            STX   VARNAM+1    and into VARNAM+1
               >40    * Skip subsequent alphanumeric characters
7EA3: 20 C4 7A >41    ]LOOP   JSR   RST100
7EA6: 90 FB    >42            BCC   ]LOOP      branch if numeric
7EA8: 20 7D E0 >43            JSR   ISLETC
7EAB: B0 F6    >44            BCS   ]LOOP      branch if alphabetic
7EAD: 90 15    >45            BCC   EXPLIC?    Always
7EAF: 4C C9 DE >46    BADNAM  JMP   SYNERR
               >47    * Code run as no explicit type specifier found, get the
               >48    * default type specifier according to 1st varname char.
7EB2: 20 5E 86 >49    SCDCH2  JSR   DECTPTR
7EB5: A6 81    >50            LDX   VARNAM

Unknown label in line: 551 >51
               >51            LDA   TYPLET-´A´,X
               >52    * Fall into implicit (2nd pass to EXPLIC?)
7EB9: 20 53 86 >53    EXPLIC? JSR   XFROMMOT   Get index from character
               >54    * No explicit type specifier found, so try implicit
               >55    * type specifier (cannot fail)
7EBC: D0 FE    >56            BNE   SCDCH2     Branch if no type spec. found

Unknown label in line: 551 >58
               >58            LDA   TVTVAL,X
7EC0: 85 11    >59            STA   VALTYP
```

```
Unknown label in line: 551 >60
                >60             LDA   TITVAL,X
7EC4: 85 12     >61             STA   INTTYP

Unknown label in line: 551 >62
                >62             LDA   TVNORA,X
7EC8: 04 81     >63             TSB   VARNAM

Unknown label in line: 551 >64
                >64             LDA   TVN1ORA,X
7ECC: 04 82     >65             TSB   VARNAM+1
7ECE: E0 02     >66             CPX   #2          FP or string
7ED0: 90 13     >67             BCC   :6
7ED2: A5 14     >68             LDA   SUBFLG
7ED4: 30 E3     >69             BMI   BADNAM
7ED6: 20 C4 7A  >70    :6       JSR   RST100      Get next character
7ED9: 38        >71             SEC
7EDA: 05 14     >72             ORA   SUBFLG
7EDC: E9 28     >73             SBC   #´(´
7EDE: D0 12     >74             BNE   :8
7EE0: 4C 9D 7F  >75    :7       JMP   NARRAY
7EE3: 24 14     >76    :8       BIT   SUBFLG
7EE5: 30 11     >77             BMI   :9
7EE7: 70 06     >78             BVS   :7
                >79    :9       DO    KOPT-K6502
7EE9: 64 14     >80             STZ   SUBFLG

Unknown label in line: 551 >85
                >85   NPTRGL90  LDX   SNCCH
7EED: F0 15     >86             BEQ   :90
7EEF: 20 57 7F  >87             JSR   SLKCACH
7EF2: D0 5E     >88             BNE   NAMFOUND    Found cache entry if Zbit clear
                >89    :90      DO    KOPT16
7EF4: 18        >90             CLC
7EF5: FB        >91             XCE
7EF6: C2 20     >92             REP   $20
7EF8: A5 69     >93             LDA   VARTAB
7EFA: 85 9B     >99    ]LOOP    STA   LOWTR
7EFC: C5 6B     >105            CMP   ARYTAB
7EFE: B0 2D     >110            BCS   NAMNTFND
7F00: B2 9B     >115            LDA   (LOWTR)
7F02: 45 81     >117            EOR   VARNAM
7F04: D0 20     >118            BNE   :1
7F06: E2 20     >129            SEP   $20         Bck to 8bits for mem/accu access
7F08: A5 12     >131            LDA   INTTYP
7F0A: 10 46     >132            BPL   NAMFOUND
7F0C: A0 06     >133            LDY   #6
7F0E: B1 9B     >134            LDA   (LOWTR),Y
7F10: 45 12     >135            EOR   INTTYP
7F12: F0 3E     >136            BEQ   NAMFOUND
7F14: C2 20     >138            REP   $20         Bck to 16bits for mem/accu
                >140   * Name not yet found: look for next variable in memory
7F16: A5 9B     >141   :1       LDA   LOWTR
                >143            MX    %01
7F18: 69 07 00  >144            ADC   #$0007      Carry already clear
7F1B: 90 DD     >145            BCC   ]LOOP       Always
```

```
                >159
7F1D: A3 01     >161  NAMNTFND LDA   1,S         16bits mem/accu. access
7F1F: 49 E9 89  >162           EOR   #RFFVL      ; upon entry to this routine
7F22: D0 15     >163           BNE   :0
7F24: 38        >164           SEC               ;Whatever the outcome..
7F25: FB        >165           XCE               ;return back to 8bits access
                >166           MX    %11
7F26: 4C 95 E0  >176           JMP   $E095       Return 0 constant
                >177  * Make new variable
7F29: 18        >178  :0       CLC
7F2A: A5 6D     >180           LDA   STREND
                >181           MX    %01
7F2C: 69 07 00  >182           ADC   #$0007
7F2F: 20 8B 7F  >183           JSR   NREASON
7F32: 20 9C E0  >192           JSR   MKNV        Make new variable (ROM routine)
7F35: A5 12     >193           LDA   INTTYP      FP or string?
7F37: 10 16     >194           BPL   :1          Yes
7F39: A0 06     >195           LDY   #6
7F3B: 91 9B     >196           STA   (LOWTR),Y
7F3D: A4 84     >197           LDY   VARPNT+1
7F3F: A5 83     >198  :1       LDA   VARPNT
7F41: 60        >199           RTS
                >200
                >201  NAMFOUND
7F42: 38        >203           SEC
7F43: FB        >204           XCE
                >205           MX    %11
7F44: 4C DE E0  >207           JMP   SETVYA
                >208
                >209  * Cache mechanism for simple variables
                >210  SCTR     EQU   LOWTR
7F47: 8A        >212  SLKCACH  TXA
7F48: 0A        >213           ASL
7F49: 85 9B     >214           STA   SCTR
7F4B: 18        >215           CLC
7F4C: FB        >216           XCE
7F4D: C2 20     >217           REP   $20
                >218           MX    %01
7F4F: A4 12     >219           LDY   INTTYP
7F51: A2 00     >220           LDX   #0
7F53: A5 81     >221           LDA   VARNAM

Unknown label in line: 551 >222
                >222  ]LOOP    CMP   SVN,X
7F57: D0 1D     >223           BNE   :0
7F59: E2 20     >224           SEP   $20         8bit access for mem/accum.
7F5B: 98        >225           TYA

Unknown label in line: 551 >226
                >226           EOR   SIT,X       ;8bit EOR operation
7F5E: C2 20     >227           REP   $20         Back to 16bits access mem/acc.
7F60: F0 1D     >228           BEQ   :1
7F62: A5 81     >229           LDA   VARNAM
7F64: E8        >230  :0       INX
7F65: E8        >231           INX
7F66: E4 9B     >232           CPX   SCTR
7F68: D0 EB     >233           BNE   ]LOOP
```

```
7F6A: 38       >234           SEC
7F6B: FB       >235           XCE
               >236           MX    %11
7F6C: 60       >255           RTS
               >256

Unknown label in line: 551 >257
               >257  :1       LDA   SLTR,X
7F6F: 85 9B    >258           STA   LOWTR
7F71: 38       >260           SEC
7F72: FB       >261           XCE
7F73: 8A       >266           TXA
7F74: 60       >267           RTS
               >268
7F75: 4C 10 D4 >269  ]ERR     JMP   MEMERR
               >271  * Pour le 65816, on entre en mode 16bits,
               >272  * Le retour se fait en mode emulation.
7F78: C5 6F    >273  NREASON  CMP   FRETOP     A/mem en mode 16bits..
7F7A: 90 1A    >274           BCC   :0
7F7C: 48       >275           PHA

Unknown label in line: 551 >276
               >276           JSR   VGARBAG
7F7F: 68       >277           PLA
7F80: C5 6F    >278           CMP   FRETOP
7F82: 08       >279  :0       PHP
7F83: 38       >280           SEC
7F84: FB       >281           XCE
               >282           MX    %11
7F85: 28       >283           PLP
7F86: B0 ED    >284           BCS   ]ERR
7F88: 60       >285           RTS
                552  * New processing for array processing
                553           PUT   PEERNARRAY
               >1    * Module handling the new array processing strategy
               >2    ERR_BSCR =     $6B
               >3    ERR_RDIM =     $78
               >4    ERR_SYNT =     $10
               >5
               >6    NUMDIM   EQU   $0F
               >7    RESULT   EQU   $62
               >8    STACK    EQU   $0100
               >9    SUBERR   EQU   $E196      Raise a BAD SUBSCRIPT error
               >10   MEMERR   EQU   $D410
               >11   REASON   EQU   $D3E3
               >12   GETARY   EQU   $E0ED
               >13   GETARY2  EQU   $E0EF      Compute addr. of 1st elm value
               >14   QINT     EQU   $EBF2
               >15
               >16   * MULTPLSS multiplies (STRNG2) by ((LOWTR),Y) leaving
               >17   * result in A,X. Hi byte also in Y
               >18   MULTPLSS EQU   $E2AD
               >19   MULTPLY1 EQU   $E2B6
               >20
7F89: 18       >22   NARRAY   CLC
7F8A: FB       >23           XCE
7F8B: C2 20    >24           REP   $20
```

```
                >25             MX    %01
7F8D: A4 14     >26             LDY   SUBFLG
7F8F: D0 51     >30             BNE   NARRGL91
7F91: A6 10     >32             LDX   DIMFLG
7F93: DA        >33             PHX
7F94: D4 11     >34             PEI   VALTYP     Pushes VALTYP/INTTYP into stack
7F96: A0 00     >43             LDY   #0
                >44   ]LOOP     MPHY
7F98: 5A        >44             PHY
7F99: D4 81     >46             PEI   VARNAM
7F9B: 38        >47             SEC
7F9C: FB        >48             XCE
                >49             MX    %11
7F9D: 20 DB 81  >56             JSR   NMAKINT
7FA0: 18        >58             CLC
7FA1: FB        >59             XCE
7FA2: C2 20     >60             REP   $20
                >61             MX    %01
7FA4: 68        >63             PLA
7FA5: 85 81     >64             STA   VARNAM     Restore array name
7FA7: 7A        >70             PLY
                >71   * Code below would transform the stack area
                >72   * from
                >73   *  DIMFLG
                >74   *  INTTYP
                >75   *  VALTYP
                >76   * SPtr ->
                >77   * to
                >78   *  (FAC+3)
                >79   *  (FAC+4)
                >80   *  DIMFLG
                >81   *  INTTYP
                >82   *  VALTYP
                >83   * SPtr ->
7FA8: A3 01     >85             LDA   1,S        Get VALTYP/INTTYP
7FAA: 48        >86             PHA
7FAB: A3 05     >87             LDA   5,S
7FAD: AA        >88             TAX              ;X set to DIMFLG
7FAE: A5 A0     >89             LDA   FAC+3      Get FAC+3,4
7FB0: EB        >90             XBA
7FB1: 83 04     >91             STA   4,S        Store it into (S+4)+5
7FB3: 38        >92             SEC
7FB4: FB        >93             XCE
                >94             MX    %11
7FB5: 8A        >95             TXA              ;Put DIMFLG back at offset
7FB6: 83 03     >96             STA   3,S        within stack frame
                >110  * Now the stack frame looks like
                >111  *  FAC+4
                >112  *  FAC+3
                >113  *  DIMFLG
                >114  *  INTTYP
                >115  *  VALTYP
                >116  * SPtr ->
7FB8: C8        >117            INY
7FB9: 20 CC 7A >118             JSR   RST102
7FBC: C9 2C     >119            CMP   #´,´
7FBE: F0 D8     >120            BEQ   ]LOOP
```

```
7FC0: 84 0F    >121           STY   NUMDIM
7FC2: 20 EF 8A >122           JSR   NCHKCLS
7FC5: 68       >123           PLA
7FC6: 85 11    >124           STA   VALTYP
7FC8: 68       >125           PLA
7FC9: 85 12    >126           STA   INTTYP
7FCB: 68       >127           PLA
7FCC: 85 10    >128           STA   DIMFLG
               >129
               >130

Unknown label in line: 553 >131
               >131  NARRGL91 LDX   ANCCH
7FD0: F0 1A    >132           BEQ   :20
7FD2: 20 FF 81 >133           JSR   ALKCACH
7FD5: D0 4C    >134           BNE   USEOLDAR
7FD7: 18       >136  :20      CLC
7FD8: FB       >137           XCE
7FD9: C2 20    >138           REP   $20
               >139           MX    %1
7FDB: A0 02    >140           LDY   #2
7FDD: A5 6B    >141           LDA   ARYTAB
7FDF: 85 9B    >142  ]LOOP    STA   LOWTR
7FE1: C5 6D    >143           CMP   STREND
7FE3: B0 39    >152           BCS   GNARRAY
7FE5: B2 9B    >154           LDA   (LOWTR)
7FE7: 45 81    >159           EOR   VARNAM
7FE9: D0 2D    >160           BNE   :5
7FEB: 38       >162           SEC
7FEC: FB       >163           XCE   Back       to mem/accu. 8bits
               >164           MX    %11
7FED: A6 12    >175           LDX   INTTYP
7FEF: 10 32    >176           BPL   USEOLDAR   If FP or string array
7FF1: 20 F7 81 >177           JSR   CNVT1
7FF4: A0 04    >178           LDY   #4
7FF6: 51 9B    >179           EOR   (LOWTR),Y
7FF8: 29 C0    >180           AND   #$C0       only test b6 and b7
7FFA: F0 27    >181           BEQ   USEOLDAR
7FFC: 18       >183           CLC
7FFD: FB       >184           XCE
7FFE: C2 20    >185           REP   $20
               >186           MX    %01
8000: A0 02    >187           LDY   #2
8002: 18       >189           CLC
               >190  :5
8003: B1 9B    >194           LDA   (LOWTR),Y
8005: 65 9B    >195           ADC   LOWTR
8007: 90 D6    >202           BCC   ]LOOP      Always
               >203
               >204  GNARRAY
8009: 38       >206           SEC
800A: FB       >207           XCE
800B: 4C 91 80 >209           JMP   MKNARRAY
               >210
800E: A5 10    >211  USEOLDAR LDA   DIMFLG     Called from the DIM stmt.?
8010: D0 7A    >212           BNE   RDIMERR
8012: A5 14    >213           LDA   SUBFLG     Subscripts given?
```

```
8014: F0 17    >214           BEQ   :1         Yes
8016: 38       >215           SEC              ;No: just return "array found"
8017: 60       >216           RTS
               >217  * Set ARYPNT to 1st elm. base addr
8018: A0 04    >218  :1       LDY   #4
801A: B1 9B    >219           LDA   (LOWTR),Y
801C: 29 07    >220           AND   #7
801E: AA       >221           TAX
801F: 20 EF E0 >222           JSR   GETARY2
8022: A5 0F    >223           LDA   NUMDIM
8024: C9 01    >224           CMP   #1
8026: F0 1C    >225           BEQ   :3
8028: E4 0F    >226           CPX   NUMDIM
802A: D0 5A    >227           BNE   SUBSERR
802C: 4C 75 81 >228           JMP   NFAEP
               >229
               >230  * Il s´agit de traiter de la reference unidimensionnelle
               >231  * sur un tableau potentiellement multi-dimensions
               >232  * Multiplier l´indice tire dans la pile par le elm size
               >233  * et comparer par rapport a l´offset du tableau (corrige
               >234  * de la taille du header).
802F: 68       >235  :3       PLA
8030: 85 AD    >236           STA   STRNG2
8032: 68       >237           PLA
8033: 85 AE    >238           STA   STRNG2+1
8035: 20 C8 81 >239           JSR   KWELMSIZ
8038: 86 64    >240           STX   RESULT+2
803A: A9 00    >241           LDA   #0
803C: 20 B6 E2 >242           JSR   MULTPLY1
803F: 86 AD    >243           STX   STRNG2
8041: 84 AE    >244           STY   STRNG2+1
8043: A0 04    >245           LDY   #4
8045: B1 9B    >246           LDA   (LOWTR),Y  # of dimensions
8047: 29 07    >247           AND   #7         Mask out new Peersoft bits
8049: 0A       >248           ASL              ;2 bytes per dimension
804A: 69 05    >249           ADC   #5         Carry clear
               >250  * Add this to element offset from base address
804C: 65 AD    >251           ADC   STRNG2
804E: A6 AE    >252           LDX   STRNG2+1
8050: 90 16    >253           BCC   :4
8052: E8       >254           INX
8053: A0 02    >255  :4       LDY   #2
8055: D1 9B    >256           CMP   (LOWTR),Y
8057: 85 83    >257           STA   VARPNT
8059: C8       >258           INY
805A: 8A       >259           TXA
805B: F1 9B    >260           SBC   (LOWTR),Y
805D: B0 27    >261           BCS   SUBSERR
805F: 86 84    >262           STX   VARPNT+1
8061: A5 9B    >263           LDA   LOWTR
8063: 65 83    >264           ADC   VARPNT
8065: 85 83    >265           STA   VARPNT
8067: A5 84    >266           LDA   VARPNT+1
8069: 65 9C    >267           ADC   LOWTR+1
806B: 85 84    >268           STA   VARPNT+1
806D: A8       >269           TAY
806E: A5 83    >270           LDA   VARPNT
```

```
8070: 60       >271           RTS
               >272
8071: A2 6B    >273  SUBSERR   LDX   #ERR_BSCR
8073: 2C       >274           HEX   2C          Skip next two bytes
8074: A2 10    >275  SNERR     LDX   #ERR_SYNT
8076: 2C       >276           HEX   2C
8077: A2 78    >277  RDIMERR   LDX   #ERR_RDIM
8079: 4C 12 D4 >278           JMP   $D412
               >279
807C: A5 14    >280  MKNARRAY  LDA   SUBFLG
807E: F0 18    >281           BEQ   :0
8080: 4C DC E1 >282           JMP   $E1DC       Raise OUT OF DATA error
8083: 20 ED E0 >283  :0        JSR   GETARY      Address 1st elm in ARYPNT&Y,A
8086: 20 C8 81 >284           JSR   KWELMSIZ
8089: 86 AD    >285           STX   STRNG2
808B: 64 BF    >287           STZ   AUXBANK
808D: A5 10    >292           LDA   DIMFLG
808F: F0 18    >293           BEQ   :1
8091: 20 55 82 >294           JSR   ISAUXMEM
8094: 18       >296  :1        CLC
8095: FB       >297           XCE
8096: C2 20    >298           REP   $20
               >299           MX    %11
8098: A5 94    >300           LDA   ARYPNT
809A: 20 8B 7F >304           JSR   NREASON     Ensure enough memory for array
809D: A5 81    >305           LDA   VARNAM
809F: 64 AE    >307           STZ   STRNG2+1
80A1: 92 9B    >308           STA   (LOWTR)
80A3: A0 01    >309           LDY   #1
80A5: A5 82    >316           LDA   VARNAM+1
80A7: 91 9B    >317           STA   (LOWTR),Y
80A9: A0 04    >318           LDY   #4
80AB: A5 12    >319           LDA   INTTYP
80AD: F0 19    >320           BEQ   :2
80AF: AA       >321           TAX
80B0: 20 F7 81 >322           JSR   CNVT1
80B3: 05 0F    >323  :2        ORA   NUMDIM
80B5: A6 BF    >324           LDX   AUXBANK
80B7: 85 BF    >325           STA   AUXBANK
80B9: 8A       >326           TXA
80BA: 0A       >327           ASL
80BB: 0A       >328           ASL
80BC: 0A       >329           ASL
80BD: 05 BF    >330           ORA   AUXBANK
80BF: 86 BF    >331           STX   AUXBANK
80C1: 91 9B    >332           STA   (LOWTR),Y
80C3: A9 00    >333  ]LOOP     LDA   #0          Hi byte of default dim
80C5: A2 0B    >334           LDX   #11         Lo byte of default dim
80C7: 24 10    >335           BIT   DIMFLG
80C9: 50 1B    >336           BVC   :5
80CB: FA       >338           PLX
80CC: 68       >339           PLA
80CD: E8       >340           INX
80CE: D0 01    >341           BNE   *+3
80D0: 1A       >342           INC
80D1: C8       >351  :5        INY               ;Add this dimension to descr.
80D2: 91 9B    >352           STA   (LOWTR),Y
```

```
80D4: C8       >353           INY
80D5: 8A       >354           TXA
80D6: 91 9B    >355           STA   (LOWTR),Y
               >356  * Multiply this dimension by running size
               >357  * ((LOWTR),Y) * (STRNG2) --> A,X
80D8: 20 AD E2 >358           JSR   MULTPLSS
80DB: 86 AD    >359           STX   STRNG2
80DD: 85 AE    >360           STA   STRNG2+1
80DF: A4 5E    >361           LDY   INDEX
80E1: C6 0F    >362           DEC   NUMDIM
80E3: D0 DE    >363           BNE   ]LOOP
               >364
80E5: A4 BF    >365           LDY   AUXBANK
80E7: F0 24    >366           BEQ   :7
80E9: A2 01    >367           LDX   #1         Ensure enough room in aux mem.
80EB: 20 2F 82 >368           JSR   ZRTAUX
80EE: E0 01    >369           CPX   #1         X set to 0 iif enough room

Bad branch in line: 553 >370
               >370           BCS   GME        otherwise -> MEMORY ERROR
80F2: A5 94    >371           LDA   ARYPNT
80F4: A4 95    >372           LDY   ARYPNT+1
80F6: 90 24    >373           BCC   :6         Always
               >374  * Now A,X has the total # of bytes of array elements
80F8: 65 95    >375  :7       ADC   ARYPNT+1   Compute address of end of array
80FA: B0 76    >376           BCS   GME        Too large: error
80FC: 85 95    >377           STA   ARYPNT+1
80FE: A8       >378           TAY
80FF: 8A       >379           TXA
8100: 65 94    >380           ADC   ARYPNT
8102: 90 18    >381           BCC   :6
8104: C8       >382           INY
8105: F0 6B    >383           BEQ   GME        Too large: error
8107: 20 E3 D3 >384  :6       JSR   REASON     Ensure enough room up to Y,A
810A: 85 6D    >385           STA   STREND
810C: 84 6E    >386           STY   STREND+1
810E: 38       >387           SEC
810F: E5 9B    >388           SBC   LOWTR
8111: A0 02    >389           LDY   #2
8113: 91 9B    >390           STA   (LOWTR),Y
8115: C8       >391           INY
8116: A5 6E    >392           LDA   STREND+1
8118: E5 9C    >393           SBC   LOWTR+1
811A: 91 9B    >394           STA   (LOWTR),Y
811C: A5 BF    >395           LDA   AUXBANK
811E: F0 3A    >396           BEQ   :9
8120: 08       >397           PHP
8121: 78       >398           SEI
8122: 8D 09 C0 >399           STA   ALTZP
8125: A5 6D    >400           LDA   STREND
8127: A6 6E    >401           LDX   STREND+1
8129: 8D 08 C0 >402           STA   STDZP
812C: 28       >403           PLP
               >404  * AUXPTR a ete fixe dans ISAUXMEM a l´adresse du slot
               >405  * Adresse du 1er element en p0.
812D: 92 06    >407           STA   (AUXPTR)
812F: A0 01    >408           LDY   #1
```

```
8131: 8A       >414           TXA
8132: 91 06    >415           STA   (AUXPTR),Y
8134: C8       >416           INY
8135: A5 AD    >417           LDA   STRNG2
8137: 91 06    >418           STA   (AUXPTR),Y
8139: C8       >419           INY
813A: A5 AE    >420           LDA   STRNG2+1
813C: 91 06    >421           STA   (AUXPTR),Y
813E: A2 02    >422           LDX   #2          Init memory slot for array
8140: 20 2F 82 >423           JSR   ZRTAUX
8143: 80 28    >424           BRA   :10
               >425  * Zero fill the element segment within the array
               >426  * (fast init).
8145: E6 AE    >427  :9        INC   STRNG2+1
8147: A4 AD    >428           LDY   STRNG2      # of byte mod 256
8149: F0 1A    >429           BEQ   :8          Upon a page limit
814B: 88       >430  ]LOOP     DEY
814C: 91 94    >431           STA   (ARYPNT),Y
814E: D0 FB    >432           BNE   ]LOOP
8150: C6 95    >433  :8        DEC   ARYPNT+1    Point to next page
8152: C6 AE    >434           DEC   STRNG2+1    Count the pages
8154: D0 F5    >435           BNE   ]LOOP       Still more to clear
8156: E6 95    >436           INC   ARYPNT+1    Rollback last Decrement
8158: A5 10    >437  :10       LDA   DIMFLG
815A: F0 19    >438           BEQ   NFAEP
815C: 60       >439           RTS
               >440
815D: 4C 10 D4 >441  GME       JMP   MEMERR      MEMORY FULL error
8160: A0 04    >442  NFAEP     LDY   #4
               >443  * New routine for ROM FIND.ARRAY.ELEMENT
               >444  * Y reg. should be 4 upon entry
8162: B1 9B    >445           LDA   (LOWTR),Y
8164: AA       >446           TAX
8165: 4A       >448           LSR
8166: 4A       >448           LSR
8167: 4A       >448           LSR
8168: 29 07    >450           AND   #7
816A: 85 BF    >451           STA   AUXBANK
816C: 8A       >452           TXA
816D: 29 07    >453           AND   #7
816F: 85 0F    >455           STA   NUMDIM
8171: A9 00    >459           LDA   #0
8173: 85 AD    >460           STA   STRNG2
8175: 85 AE    >461  ]LOOP     STA   STRNG2+1
8177: C8       >462           INY               ;Pull next subscript from stack
8178: FA       >463           PLX
8179: 86 A0    >464           STX   FAC+3
817B: 68       >465           PLA
817C: 85 A1    >466           STA   FAC+4
817E: D1 9B    >467           CMP   (LOWTR),Y
8180: 90 20    >468           BCC   FAE2
8182: D0 1B    >469           BNE   GSE         Subscript is too large
8184: C8       >470           INY
8185: 8A       >471           TXA
8186: D1 9B    >472           CMP   (LOWTR),Y
8188: 90 19    >473           BCC   FAE3
818A: 4C 96 E1 >474  GSE       JMP   SUBERR      BAD SUBSCRIPT error
```

```
818D: C8       >475  FAE2     INY
818E: A5 AE    >476  FAE3     LDA   STRNG2+1   Bypass multiplication if
8190: 05 AD    >477          ORA   STRNG2      value so far is zero
8192: 18       >478          CLC
8193: F0 1F    >479          BEQ   :1
8195: 20 AD E2 >480          JSR   MULTPLSS
8198: 8A       >481          TXA                ;Add current subscript
8199: 65 A0    >482          ADC   FAC+3
819B: AA       >483          TAX
819C: 98       >484          TYA
819D: A4 5E    >485          LDY   INDEX
819F: 65 A1    >486  :1       ADC   FAC+4       Finish adding current subscript
81A1: 86 AD    >487          STX   STRNG2      Store accumulated offset
81A3: C6 0F    >488          DEC   NUMDIM      Last subscript yet?
81A5: D0 CE    >489          BNE   ]LOOP       No: loop till done
81A7: 85 AE    >490          STA   STRNG2+1    Yes: multiply by element size
81A9: 20 C8 81 >491          JSR   KWELMSIZ
81AC: A5 BF    >492          LDA   AUXBANK
81AE: F0 15    >493          BEQ   :2
81B0: 4C 98 E2 >494  :2       JMP   $E298
               >495
               >496  * Donne la taille de l´element en fonction
               >497  * de VARNAM,+1 et de INTTYP
               >498  * Result in X reg.
81B3: 24 81    >499  KWELMSIZ BIT   VARNAM
81B5: 10 1B    >500          BPL   :0
81B7: A5 12    >501          LDA   INTTYP
81B9: 29 07    >502          AND   #7
81BB: AA       >503          TAX
81BC: 60       >504          RTS
81BD: A2 05    >505  :0       LDX   #5
81BF: 24 82    >506          BIT   VARNAM+1
81C1: 10 17    >507          BPL   :1
81C3: CA       >508          DEX                ;Back to 3 if string
81C4: CA       >509          DEX
81C5: 60       >510  :1       RTS
               >511
               >512  * Evaluate numeric formula at TXTPPTR
               >513  * Converting result to INTEGER 0<= X < 65536
               >514  * into FAC+3,4
81C6: 20 C4 7A >515  NMAKINT  JSR   RST100     Get next character
81C9: 20 8C 89 >516          JSR   NFRMNUM
               >517  * Convert FAC to integer
81CC: A5 A2    >518          LDA   FACSIGN
81CE: 30 24    >519          BMI   :1
81D0: A5 9D    >520          LDA   FAC
81D2: C9 90    >521          CMP   #$90
81D4: 90 1B    >522          BCC   :3          Branch if abs(value) < 32768
81D6: 20 C4 92 >523          JSR   GN65536
81D9: 20 BE E7 >524          JSR   FADD
81DC: 4C F2 EB >525  :3       JMP   QINT
81DF: 4C 99 E1 >526  :1       JMP   GOIQERR
               >527
               >528  * Convert INTTYP (in X reg.) from $81 to $84
               >529  * to %0000_0000 to %1100_0000 (respectively)
               >530  * Output value could be ORA ed or EOR ed with
               >531  * NUMDIM slot with array structure
```

```
81E2: CA       >532  CNVT1    DEX
81E3: 8A       >533           TXA
81E4: 4A       >534           LSR              ;b0 into Carry, 0 into b7
81E5: 6A       >535           ROR              ;b0 into b7 and b1 into carry
81E6: 6A       >536           ROR              ;b0 into b6, b1 into b7
81E7: 29 C0    >537           AND   #$C0       Only retain b6-b7
81E9: 60       >538           RTS
               >539
               >540  * Cache mechanism for array variables
               >541  ACTR     EQU   LOWTR
81EA: 8A       >543  ALKCACH  TXA
81EB: 0A       >544           ASL
81EC: 85 9B    >545           STA   SCTR
81EE: 18       >546           CLC
81EF: FB       >547           XCE
81F0: C2 20    >548           REP   $20
               >549           MX    %10
81F2: A4 12    >550           LDY   INTTYP
81F4: A2 00 00 >551           LDX   #0

Unknown label in line: 553 >552
               >552  ]LOOP    CMP   AVN,X
81F9: D0 22    >553           BNE   :0
81FB: E2 20    >554           SEP   $20
               >555           MX    %11
81FD: 98       >556           TYA

Unknown label in line: 553 >557
               >557           EOR   AIT,X      8bit EOR operation
8200: C2 20    >558           REP   $20
               >559           MX    %10
8202: F0 22    >560           BEQ   :1
8204: A5 81    >561           LDA   VARNAM
8206: E8       >562  :0       INX
8207: E8       >563           INX
8208: E4 9B    >564           CPX   SCTR
820A: D0 EB    >565           BNE   ]LOOP
820C: 38       >566           SEC
820D: FB       >567           XCE
               >568           MX    %11
820E: 60       >587           RTS
               >588

Unknown label in line: 553 >589
               >589  :1       LDA   ALTR,X
8211: 85 9B    >590           STA   LOWTR
8213: 38       >592           SEC
8214: FB       >593           XCE
8215: 8A       >598           TXA
8216: 60       >599           RTS
               >600
               >601  * Common entry point for accessing array content
               >602  * within auxiliary memory.
8217: A9 BF    >603  ZRTAUX   LDA   #$BF
8219: 8D EE 03 >604           STA   $03EE
821C: 9C ED 03 >606           STZ   $03ED
821F: B8       >611           CLV
```

```
8220: 38       >612           SEC
8221: 4C 14 C3 >613           JMP   XFER
               >614
8224: 2C 83 C0 >615  NGARBAG  BIT   $C083
8227: 2C 83 C0 >616           BIT   $C083
822A: 20 00 D0 >617           JSR   $D000
822D: 2C 81 C0 >618           BIT   $C081
8230: 2C 81 C0 >619           BIT   $C081
8233: 60       >620           RTS
                554  * New strategy for array storage
                555           PUT   PEERNAUXMEM
               >1    * Module handling the new Peersoft array storage strategy
               >2
8234: 4C C9 DE >3    GSNERR2  JMP   SYNERR
8237: 4C 99 E1 >4    GIQERR2  JMP   GOIQERR
823A: 4C 76 DD >5    GTMERR2  JMP   GOTMIERR
               >6    * Routine to test whether the array will be located
               >7    * Outcome:
               >8    * Carry set iif aux. mem storage asked for
               >9    * AUXBANK: bank memory asked for (in bits b4..b5)
               >10   * ARYPNT,+1: incremented if aux mem. storage
               >11   * (placeholders for offset within aux memory and
               >12   *  one element of specified size for returning values
               >13   *  during value expressions
               >14   * Y,A: values incremented in case aux. mem storage
823D: B2 B8    >16   ISAUXMEM LDA   (TXTPTR)
823F: C9 23    >20           CMP   #´#´
8241: 18       >21           CLC
8242: D0 4F    >22           BNE   :2
8244: 20 C4 7A >23           JSR   RST100     Next char. must be numeric
8247: B0 03    >24           BCS   GSNERR2    otherwise SYNTAX ERROR
8249: 29 07    >25           AND   #7
               >26   * Pour le moment uniquement la memoire auxiliaire
               >27   * est autorisee
824B: C9 02    >28           CMP   #2
824D: B0 00    >29           BCS   GIQERR2
824F: 85 BF    >30           STA   AUXBANK
8251: 20 C4 7A >31           JSR   RST100     Point to next character
8254: 18       >32           CLC
               >33   * test de conformance par rap. a la configuration hote

Unknown label in line: 555 >34
               >34           BIT   MEMORY     b6 a 1 si carte mem aux.
8257: A9 01    >36           LDA   #1
8259: 50 01    >37           BVC   *+3
825B: 3A       >38           DEC
825C: 14 BF    >39           TRB   AUXBANK
825E: A5 BF    >40           LDA   AUXBANK
8260: F0 31    >49           BEQ   :2
8262: A5 94    >50           LDA   ARYPNT
8264: A4 95    >51           LDY   ARYPNT+1
8266: 85 06    >52           STA   AUXPTR
8268: 84 07    >53           STY   AUXPTR+1
826A: 65 AD    >54           ADC   STRNG2     Carry already clear
826C: 90 02    >55           BCC   *+4
826E: C8       >56           INY
826F: 18       >57           CLC
```

```
8270: 69 04    >58            ADC   #4
8272: 90 01    >59            BCC   *+3
8274: C8       >60            INY
8275: 84 95    >61            STY   ARYPNT+1
8277: 38       >62            SEC
8278: 85 94    >63            STA   ARYPNT
827A: A5 94    >64   :2       LDA   ARYPNT
827C: 60       >65   ]LOOP    RTS
               >66

Unknown label in line: 555 >67
               >67   RCLMAUX  BIT   MEMORY
827F: 50 FB    >68            BVC   ]LOOP
8281: A2 00    >69            LDX   #0         Init array storage in aux mem.
8283: 4C 2F 82 >70            JMP   ZRTAUX
                556
                557  * Upon init, all variables are floating point by default
8286: 08        558  LBS00    PHP
8287: A2 1A     559            LDX   #26
8289: A9 21     560            LDA   #´!´

Unknown label in line: 561
                561  ]LOOP    STA   TYPLET-1,X
828D: CA        562            DEX
828E: D0 FB     563            BNE   ]LOOP

Unknown label in line: 564
                564            STX   AEI

Unknown label in line: 565
                565            STX   AEI+1
                566  * Reinit variables lookup caches (simple & array)

Unknown label in line: 567
                567            STX   SNCCH

Unknown label in line: 568
                568            STX   ANCCH

Unknown label in line: 569
                569            STX   WMODE
829A: 20 96 82  570            JSR   RCLMAUX
829D: 28        571            PLP
829E: 60        572            RTS
                573
                574  * Applesoft RUN command
829F: 20 A0 82  575  RRUN      JSR   LBS00      Init the default vartype table

Unknown label in line: 576
                576            STX   MONU       Rearms MOUSE instruction flag
82A4: 4C 12 D9  577            JMP   $D912
                578
                579  * Applesoft NEW command
82A7: 20 A0 82  580  RNEW      JSR   LBS00
82AA: 4C 4B D6  581            JMP   $D64B
                582
                583  * Applesoft CLEAR command
```

```
82AD: 20 A0 82  584  RCLEAR   JSR   LBS00
82B0: 4C 6C D6  585           JMP   $D66C
                586
82B3: 20 7D E0  587  MISLETC  JSR   ISLETC
82B6: 90 29     588           BCC   GOSYNERR
82B8: 60        589           RTS
                590
                591  * New subroutine checking a character (code in A)
                592  * is pointed to by TXTPTR
                593  * Falls into SYNERR if not
                594  NSYNCHR  DO    KOPT-K65C02
82B9: D2 B8     598  NSYNCHR2 CMP   (TXTPTR)
82BB: D0 24     600           BNE   GOSYNERR
82BD: 4C C4 7A  601           JMP   RST100
82C0: 4C C9 DE  602  GOSYNERR JMP   SYNERR
                603
                604           PUT   PEERPROCFUN
               >1    * Module en charge des fonctions utilisateur
               >2    * et particulierement des PF
               >3    ARG      EQU   $A5
               >4    TRCFLG   EQU   $F2
               >5    BISVTYP  EQU   $BE
               >6    VECTUSR  EQU   $A
               >7    TMERR    EQU   $DD76
               >8    ULERR    EQU   $D97C
               >9    MOVFM    EQU   $EAF9
               >10   MOVFA    EQU   $EB53
               >11   LET2     EQU   $DA63
               >12
               >13            DUMMY 0
0000: 00       >14   USRMOD   DS    1
0001: 00 00    >15   ADRUSR   DS    2
0003: 00 00    >16   VSRTNAM  DS    2
0005: 00       >17   VSRTVT   DS    1
0006: 00       >18   VSRTIT   DS    1
0007: 00 00    >19   VSRTPTR  DS    2
0009: 00 00    >20   VENT1NAM DS    2
000B: 00       >21   VENT1VT  DS    1
000C: 00       >22   VENT1IT  DS    1
000D: 00 00    >23   VENT1PTR DS    2
000F: 00 00    >24   VENT2NAM DS    2
0011: 00       >25   VENT2VT  DS    1
0012: 00       >26   VENT2IT  DS    1
0013: 00 00    >27   VENT2PTR DS    2
               >28   LENREC   EQU   *
               >29            DEND
               >30   * Sous routine pour initialiser les routines USR de type
               >31   * PF.
82C3: A2 0A    >32   RAZPF    LDX   #10
               >33   ]LOOP    MPHX
82C5: DA       >33            PHX
82C6: 20 0D 83 >34            JSR   COMPOFST
82C9: FA       >35            PLX
82CA: B2 06    >37            LDA   (AUXPTR)
82CC: 10 27    >42            BPL   :0
82CE: A0 02    >43            LDY   #ADRUSR+1
82D0: A9 00    >44            LDA   #0
```

```
82D2: 91 06    >45             STA   (AUXPTR),Y
82D4: CA       >46    :0       DEX
82D5: 10 EE    >47             BPL   ]LOOP

Unknown label in line: 604 >48
               >48             STX   PFINDIC

Unknown label in line: 604 >50
               >50             STZ   ISPFACT
82DB: 60       >55             RTS
               >56
82DC: A2 0B    >57   SETINITX LDX   #12-1

Unknown label in line: 604 >58
               >58   ]LOOP     LDA   SINITX,X
82E0: 95 69    >59             STA   $69,X

Unknown label in line: 604 >60
               >60             STA   SVALTNM,X
82E4: CA       >61             DEX
82E5: 10 F7    >62             BPL   ]LOOP
82E7: 60       >63             RTS
               >64
               >65   * Indice de la fonction dans X, ramene dans A,Y
               >66   * L´adresse de debut de la structure
82E8: A9 00    >67   COMPOFST LDA   #0
82EA: A8       >68             TAY
82EB: F0 2A    >69             BEQ   :00        Always
82ED: 69 15    >70   ]LOOP     ADC   #LENREC
82EF: 90 27    >71             BCC   :0
82F1: C8       >72             INY
82F2: 18       >73   :00       CLC
82F3: CA       >74   :0        DEX
82F4: 10 F7    >75             BPL   ]LOOP
82F6: 69
Unknown label in line: 604 >76
               >76             ADC   #ADRSTRUCT
82F9: 48       >77             PHA
82FA: 98       >78             TYA
82FB: 69
Unknown label in line: 604 >79
               >79             ADC   #>ADRSTRUCT
82FE: A8       >80             TAY
82FF: 68       >81             PLA
8300: 85 06    >82             STA   AUXPTR
8302: 84 07    >83             STY   AUXPTR+1
8304: 60       >84             RTS
               >85
8305: 18       >86   GOSVCUR  CLC
               >87   ]LOOP
               >88   * Connaitre tout d´une variable non encore enregistree
               >89   * A: offset du premier byte pour la var. dans structure
8306: 4C 76 DD >90   ]ERR      JMP   TMERR
8309: 48       >91   FRSTIM   PHA
830A: 20 F2 8A >92             JSR   NCHKCOM
830D: B2 06    >94             LDA   (AUXPTR)
830F: 29 01    >99             AND   #1         Environnement dynamique oui/non
```

```
8311: 48       >100           PHA
8312: F0 32    >101           BEQ   :0
8314: A2 0B    >102           LDX   #12-1
8316: B5 69    >103  ]LOOP    LDA   $69,X

Unknown label in line: 604 >104
               >104           STA   SVCURRM,X

Unknown label in line: 604 >105
               >105           LDA   SDEF1,X
831C: 95 69    >106           STA   $69,X
831E: CA       >107           DEX
831F: 10 F5    >108           BPL   ]LOOP
8321: D4 06    >110  :0       PEI   AUXPTR
8323: 20 8E 7E >117           JSR   NPTRGTX
8326: C5 6B    >118           CMP   ARYTAB
8328: 98       >119           TYA
8329: E5 6C    >120           SBC   ARYTAB+1
832B: 68       >121           PLA
832C: 85 06    >122           STA   AUXPTR
832E: 68       >123           PLA
832F: 85 07    >124           STA   AUXPTR+1
8331: 68       >125           PLA
8332: F0 2F    >126           BEQ   :1
8334: A2 0B    >127           LDX   #12-1

Unknown label in line: 604 >128
               >128  ]LOOP    LDA   SVCURRM,X
8338: 95 69    >129           STA   $69,X
833A: CA       >130           DEX
833B: 10 F9    >131           BPL   ]LOOP
833D: B0 C7    >132  :1       BCS   ]ERR
833F: 7A       >133           PLY
8340: A5 81    >134           LDA   VARNAM
8342: 91 06    >135           STA   (AUXPTR),Y
8344: C8       >136           INY
8345: A5 82    >137           LDA   VARNAM+1
8347: 91 06    >138           STA   (AUXPTR),Y
8349: C8       >139           INY
834A: A5 11    >140           LDA   VALTYP
834C: 91 06    >141           STA   (AUXPTR),Y
834E: C8       >142           INY
834F: A5 12    >143           LDA   INTTYP
8351: 91 06    >144           STA   (AUXPTR),Y
8353: C8       >145           INY
8354: A5 83    >146  COMX1     LDA   VARPNT
8356: 91 06    >147           STA   (AUXPTR),Y
8358: C8       >148           INY
8359: A5 84    >149           LDA   VARPNT+1
835B: 91 06    >150           STA   (AUXPTR),Y
835D: 60       >151           RTS
               >152
               >153  * Connaitre tout d´une variable deja enregistree
               >154  * Y offset dans structure... (adressage par
               >155  * (AUXPTR),Y
835E: B1 06    >156  SCNDTIM   LDA   (AUXPTR),Y
8360: 85 81    >157           STA   VARNAM
```

```
8362: C8       >158           INY
8363: B1 06    >159           LDA   (AUXPTR),Y
8365: 85 82    >160           STA   VARNAM+1
8367: C8       >161           INY
8368: B1 06    >162           LDA   (AUXPTR),Y
836A: 85 11    >163           STA   VALTYP
836C: C8       >164           INY
836D: B1 06    >165           LDA   (AUXPTR),Y
836F: 85 12    >166           STA   INTTYP
8371: C8       >167           INY
8372: 5A       >168           PHY
8373: 20 FA 7E >169           JSR   NPTRGL90
8376: 7A       >170           PLY
8377: 80 01    >171           BRA   COMX1
               >172
               >173  * X,A adresse a sauver dans ADRUSR de la structure
8379: A0 01    >174  HNDLEADR LDY   #ADRUSR
837B: 91 06    >175           STA   (AUXPTR),Y
837D: 90 2E    >176           BCC   :4
837F: 85 0B    >177           STA   $0B
8381: 86 0C    >178           STX   $0C
8383: A9 4C    >179           LDA   #$4C
8385: 85 0A    >180           STA   $0A
8387: C8       >181  :4        INY
8388: 8A       >182           TXA
8389: 91 06    >183           STA   (AUXPTR),Y
838B: 60       >184           RTS
               >185
838C: B1 06    >186  COMLET2  LDA   (AUXPTR),Y
838E: AA       >187           TAX              ;INTTYP dans X
838F: C8       >188           INY
8390: B1 06    >189           LDA   (AUXPTR),Y ;pointeur sur valeur
8392: 85 85    >190           STA   FORPNT      dans FORPNT
8394: C8       >191           INY
8395: B1 06    >192           LDA   (AUXPTR),Y
8397: 85 86    >193           STA   FORPNT+1
8399: 8A       >194           TXA              ;Set bit N
839A: 4C 63 DA >195           JMP   LET2
               >196
839D: 4C 10 D4 >197  ]ERR      JMP   MEMERR
83A0: 20 C4 7A >198  RUSR      JSR   RST100
83A3: A2 0A    >199           LDX   #10
83A5: B0 2C    >200           BCS   :0         Not a digit
83A7: E9 2F    >201           SBC   #´0´-1
83A9: AA       >202           TAX
83AA: 20 C4 7A >203           JSR   RST100
               >204  :0        MPHX
83AD: DA       >204           PHX
83AE: 20 0D 83 >205           JSR   COMPOFST
83B1: B2 06    >207           LDA   (AUXPTR)
83B3: 29 40    >212           AND   #64
83B5: F0 67    >213           BEQ   :1
83B7: BA       >214           TSX
83B8: E0 08    >215           CPX   #8         At least 8 bytes on stack OK
83BA: 90 E1    >216           BCC   ]ERR
83BC: 20 F5 8A >217           JSR   NCHKOPN
83BF: 20 7B DD >218           JSR   FRMEVL
```

```
83C2: BA       >219           TSX
83C3: A5 11    >220           LDA   VALTYP
83C5: 9D 00 01 >221           STA   $0100,X
83C8: 8A       >222           TXA
83C9: 38       >223           SEC
83CA: E9 06    >224           SBC   #6
83CC: AA       >225           TAX
83CD: 9A       >226           TXS
83CE: E8       >227           INX
83CF: A0 01    >228           LDY   #1
83D1: 20 2B EB >229           JSR   MOVMF
83D4: 20 F2 8A >230           JSR   NCHKCOM
83D7: 20 EC 8A >231           JSR   NPARCHK+3  2nd arg value left in FAC
83DA: BA       >232           TSX
83DB: E8       >233           INX
83DC: 8A       >234           TXA
83DD: 48       >235           PHA
83DE: A0 01    >236           LDY   #1
83E0: 20 E3 E9 >237           JSR   $E9E3      Load ARG from Y,A/1st arg value
83E3: 68       >238           PLA
83E4: 18       >239           CLC
83E5: 69 05    >240           ADC   #5         6 instead of 5 because of INX
83E7: AA       >241           TAX
83E8: BD 00 01 >242           LDA   $0100,X
83EB: 85 BE    >243           STA   BISVTYP
83ED: 9A       >244           TXS
83EE: 80 31    >245           BRA   :2
83F0: A2 26    >246  ]ERR     LDX   #38
83F2: 2C       >247           HEX   2C         Skip next two bytes
83F3: A2 27    >248  ]ERR1    LDX   #39
83F5: 4C 50 92 >249           JMP   NERRH
83F8: 20 E9 8A >250  :1       JSR   NPARCHK    1er ou 2eme parm dans FAC
               >251  :2       MPLX
83FB: FA       >251           PLX
83FC: DA       >253           PHX
83FD: 20 0D 83 >257           JSR   COMPOFST   Set AUXPTR according index X
8400: A0 02    >258           LDY   #ADRUSR+1
8402: B1 06    >259           LDA   (AUXPTR),Y
8404: F0 EA    >260           BEQ   ]ERR
8406: FA       >261           PLX

Unknown label in line: 604 >262
               >262           STX   PFINDX
8409: B2 06    >264           LDA   (AUXPTR)
840B: 10 6C    >269           BPL   V3
               >270  * Procedural function...
840D: 4A       >271           LSR
840E: 90 51    >272           BCC   :10        Branchem. ssi pas de segment

Unknown label in line: 604 >273
               >273           LDA   ISPFACT
8412: D0 DF    >274           BNE   ]ERR1
8414: DA       >275           PHX
8415: 20 B4 85 >276           JSR   SAVCURRM
8418: 68       >277           PLA

Unknown label in line: 604 >278
```

```
                >278            CMP   PFINDIC
841B: F0 2C     >279            BEQ   :11
841D: 20 FF 82  >280            JSR   SETINITX
8420: 20 A9 85  >281  :11       JSR   RSTALTM
8423: A0 03     >282            LDY   #VSRTNAM
8425: 20 84 83  >283            JSR   SCNDTIM
8428: A0 09     >284            LDY   #VENT1NAM
842A: 20 84 83  >285            JSR   SCNDTIM
842D: B2 06     >287            LDA   (AUXPTR)
842F: 29 40     >292            AND   #64
8431: F0 2E     >293            BEQ   :10
8433: A0 0F     >294            LDY   #VENT2NAM
8435: 20 84 83  >295            JSR   SCNDTIM
8438: A0 0C     >296  :10       LDY   #VENT1IT
843A: 20 B2 83  >297            JSR   COMLET2
843D: B2 06     >299            LDA   (AUXPTR)
843F: 29 40     >304            AND   #64
8441: F0 31     >305            BEQ   :12
8443: 20 53 EB  >306            JSR   MOVFA
8446: A0 12     >307            LDY   #VENT2IT
8448: 20 B2 83  >308            JSR   COMLET2
                >309  :12       DO    KOPT16
844B: F4 47 85  >310            PEA   RETOUR-1
844E: 80 3F     >317            BRA   COMMONG
                >318
                >319  * Code run when parsing USR function that is not a PF
8450: E0 0A     >320  V3        CPX   #10
8452: B0 38     >321            BCS   :4          Special case for original USR
8454: A0 02     >322            LDY   #ADRUSR+1
8456: B1 06     >323            LDA   (AUXPTR),Y
8458: AA        >324            TAX
8459: 88        >325            DEY
845A: B1 06     >326            LDA   (AUXPTR),Y
845C: D0 01     >327            BNE   *+3
845E: CA        >328            DEX
845F: 3A        >330            DEC
8460: DA        >336            PHX
8461: 48        >337            PHA
8462: 60        >343            RTS
8463: 4C 0A 00  >344  :4        JMP   VECTUSR
                >345
8466: A0
Unknown label in line: 604 >346
                >346  COMMONG   LDY   #FINOF-SVOFST-1

Unknown label in line: 604 >347
                >347  ]LOOP     LDX   SVOFST,Y
846B: B5 00     >348            LDA   0,X

Unknown label in line: 604 >349
                >349            STA   SVAREA,Y
846F: 88        >350            DEY
8470: 10 F7     >351            BPL   ]LOOP
8472: 64 F2     >353            STZ   TRCFLG
                >358  * This is the critical code segment
8474: D4 B8     >360            PEI   TXTPTR
8476: D4 75     >361            PEI   CURLIN
```

```
8478: A9 B0    >372           LDA   #TOKGOSUB
847A: 48       >373           PHA
847B: A0 01    >374           LDY   #ADRUSR
847D: B1 06    >375           LDA   (AUXPTR),Y
847F: 85 B8    >376           STA   TXTPTR
8481: C8       >377           INY
8482: B1 06    >378           LDA   (AUXPTR),Y
8484: 85 B9    >379           STA   TXTPTR+1
8486: 4C D2 D7 >380           JMP   NEWSTT
               >381
8489: 20 CC 7A >382  RDEFUSR  JSR   RST102
848C: 90 2F    >383           BCC   :1          Branch if digit
848E: A9 0A    >384           LDA   #10
8490: 48       >385           PHA
8491: D0 30    >386           BNE   :3          Always
8493: E9 2F    >387  :1       SBC   #´0´-1      ASCII digit to binary
8495: 48       >388           PHA
8496: 20 C4 7A >389           JSR   RST100
8499: A9 D0    >390  :3       LDA   #TOKEQUAL
849B: 20 DA 82 >391           JSR   NSYNCHR
849E: 20 67 DD >392           JSR   FRMNUM
84A1: 20 52 E7 >393           JSR   GETADR
84A4: FA       >394           PLX
84A5: DA       >396           PHX
84A6: 20 0D 83 >400           JSR   COMPOFST
84A9: 68       >401           PLA
84AA: 48       >402           PHA
84AB: C9 0A    >403           CMP   #10         Set carry flag
               >404  * If LINNUM high byte is zero, then must be the mode
84AD: A5 50    >405           LDA   LINNUM
84AF: A6 51    >406           LDX   LINNUM+1
84B1: F0 3B    >407           BEQ   :5
84B3: 20 9F 83 >408           JSR   HNDLEADR
84B6: 68       >409           PLA
84B7: A9 00    >410           LDA   #0
84B9: 92 06    >412           STA   (AUXPTR)
84BB: 20 CC 7A >417  ]LOOP    JSR   RST102
84BE: D0 01    >418           BNE   *+3
84C0: 60       >419           RTS
84C1: 4C C9 DE >420  ]ERR     JMP   SYNERR
               >421  * DEFUSR=<mode>,<otherparms>
84C4: 92 06    >423  :5       STA   (AUXPTR)
84C6: A8       >428           TAY
84C7: 30 4E    >429           BMI   :6          Procedural function
84C9: 29 3F    >430           AND   #$3F
84CB: D0 F4    >431           BNE   ]ERR
84CD: 20 F2 8A >432           JSR   NCHKCOM
84D0: 20 67 DD >433           JSR   FRMNUM
84D3: 20 52 E7 >434           JSR   GETADR
84D6: FA       >435           PLX
84D7: E0 0A    >436           CPX   #10
84D9: 08       >437           PHP
84DA: 20 0D 83 >438           JSR   COMPOFST
84DD: 28       >439           PLP
84DE: A5 50    >440           LDA   LINNUM
84E0: A6 51    >441           LDX   LINNUM+1
84E2: 4C 9F 83 >442  ]LOOP    JMP   HNDLEADR
```

```
84E5: 4C 7C D9 >443  ]ERR     JMP   ULERR
84E8: A2 28    >444  ]ERR1    LDX   #40
84EA: 4C 50 92 >445          JMP   NERRH
84ED: 48       >446  :6      PHA

Unknown label in line: 604 >447
               >447          LDA   ISPFACT
84F0: D0 F6    >448          BNE   ]ERR1
84F2: A9 03    >449          LDA   #VSRTNAM
84F4: 20 2C 83 >450          JSR   FRSTIM
84F7: A9 09    >451          LDA   #VENT1NAM
84F9: 20 2C 83 >452          JSR   FRSTIM
84FC: 68       >453          PLA
84FD: 29 40    >454          AND   #64
84FF: F0 30    >455          BEQ   :7
8501: A9 0F    >456          LDA   #VENT2NAM
8503: 20 2C 83 >457          JSR   FRSTIM
8506: 68       >458  :7      PLA             ;Do not care routine idx
8507: 20 F2 8A >459          JSR   NCHKCOM
850A: 20 0C DA >460          JSR   LINGET
850D: 20 1A D6 >461          JSR   FNDLIN
8510: 90 D3    >462          BCC   ]ERR
8512: A6 9C    >463          LDX   LOWTR+1
8514: A5 9B    >464          LDA   LOWTR
8516: D0 01    >465          BNE   *+3
8518: CA       >466          DEX
8519: 3A       >468          DEC
851A: 18       >472          CLC
851B: 90 C5    >473          BCC   ]LOOP     Always
               >474
851D: 20 68 85 >475  RETOUR   JSR   COMREST

Unknown label in line: 604 >476
               >476          LDX   PFINDX
8522: DA       >477          PHX
8523: 20 0D 83 >478          JSR   COMPOFST
8526: 20 76 85 >479          JSR   COLLECTR
8529: FA       >480          PLX
852A: B2 06    >482          LDA   (AUXPTR)

Unknown label in line: 604 >483
               >483          STZ   ISPFACT
852E: 4A       >489          LSR
852F: 90 36    >490          BCC   :0

Unknown label in line: 604 >491
               >491          STX   PFINDIC
8533: 20 BF 85 >492          JSR   SAVALTM
8536: 4C 9E 85 >493          JMP   RSTCURRM
8539: 60       >494  :0      RTS
               >495
853A: A0
Unknown label in line: 604 >496
               >496  COMREST  LDY   #FINOF-SVOFST-1

Unknown label in line: 604 >497
               >497  ]LOOP    LDX   SVOFST,Y
```

```
Unknown label in line: 604 >498
                >498            LDA   SVAREA,Y
8541: 95 00     >499            STA   0,X
8543: 88        >500            DEY
8544: 10 F7     >501            BPL   ]LOOP
8546: 60        >502            RTS
                >503
8547: A0 06     >504  COLLECTR  LDY   #VSRTIT
8549: B1 06     >505            LDA   (AUXPTR),Y
854B: 0A        >506            ASL
854C: A0 07     >507            LDY   #VSRTPTR
854E: B1 06     >508            LDA   (AUXPTR),Y
8550: AA        >509            TAX
8551: C8        >510            INY
8552: B1 06     >511            LDA   (AUXPTR),Y
8554: A8        >512            TAY
8555: 8A        >513            TXA
8556: B0 36     >514            BCS   :0          Branch iif integer output var.
8558: 64 11     >516            STZ   VALTYP
855A: 64 12     >517            STZ   INTTYP
855C: 4C F9 EA >523             JMP   MOVFM
855F: 84 84     >524  :0        STY   VARPNT+1
8561: 85 83     >525            STA   VARPNT
8563: B2 83     >527            LDA   (VARPNT)
8565: A0 01     >528            LDY   #1
8567: AA        >534            TAX
8568: B1 83     >535            LDA   (VARPNT),Y
856A: A8        >536            TAY
856B: 8A        >537            TXA
856C: 4C F2 E2 >538             JMP   GIVAYF
                >539
856F: A2 0B     >540  RSTCURRM  LDX   #12-1

Unknown label in line: 604 >541
                >541  ]LOOP     LDA   SVCURRM,X
8573: 95 69     >542            STA   $69,X
8575: CA        >543            DEX
8576: 10 F9     >544            BPL   ]LOOP
8578: 60        >545            RTS
                >546
8579: A2 0B     >547  RSTALTM   LDX   #12-1

Unknown label in line: 604 >548
                >548  ]LOOP     LDA   SVALTNM,X
857D: 95 69     >549            STA   $69,X
857F: CA        >550            DEX
8580: 10 F9     >551            BPL   ]LOOP
8582: 60        >552            RTS
                >553
8583: A2 0B     >554  SAVCURRM  LDX   #12-1
8585: B5 69     >555  ]LOOP     LDA   $69,X

Unknown label in line: 604 >556
                >556            STA   SVCURRM,X
8589: CA        >557            DEX
858A: 10 F9     >558            BPL   ]LOOP
```

```
858C: 60       >559           RTS
               >560
858D: A2 0B    >561  SAVALTM  LDX   #12-1
858F: B5 69    >562  ]LOOP    LDA   $69,X

Unknown label in line: 604 >563
               >563           STA   SVALTNM,X
8593: CA       >564           DEX
8594: 10 F9    >565           BPL   ]LOOP
8596: 60       >566           RTS
                605           PUT   PEERDEF
               >1    * Nouvelle routine de traitement du DEF..
8597: 4C B3 84 >2    ]LOOP    JMP   RDEFUSR
859A: A4 B9    >3    RDEF     LDY   TXTPTR+1
859C: A5 B8    >4             LDA   TXTPTR
859E: D0 01    >11            BNE   *+3
85A0: 88       >12            DEY
85A1: 3A       >13            DEC
85A2: A2 01    >15            LDX   #1
85A4: 20 19 87 >16            JSR   RECON      Check which DEF pattern
85A7: D0 36    >17            BNE   :1         None detected
85A9: 4C 13 E3 >18            JMP   $E313
85AC: 88       >19   :1       DEY
85AD: 20 98 D9 >20            JSR   ADDON
85B0: A6 BD    >21            LDX   IDMOCL
85B2: E0
Unknown label in line: 605 >22
               >22            CPX   #OFFUSR-TOFFST Is it DEFUSR?
85B5: F0 E0    >23            BEQ   ]LOOP

Unknown label in line: 605 >24
               >24            LDA   MOTIF-NOPER-7,X Must be DEF(INT/STR/SNG)
               >25   * Below is the common code for all three new instructions
85B9: 64 C0    >30            STZ   LETINF
85BB: 85 C1    >32            STA   TYPMOD
85BD: 20 5E 86 >33            JSR   DECTPTR    Decrement TXTPTR
85C0: 20 29 86 >34   ]LOOP    JSR   :LBS00     Bump ptr. to 1st letter of next v
ar
85C3: 20 D4 82 >35            JSR   MISLETC    Must be alphabetic
85C6: 85 C0    >36            STA   LETINF
85C8: 20 29 86 >37            JSR   :LBS00     Exit if no further variable
85CB: C9 C9    >38            CMP   #TOKMINUS  means a letter range
85CD: F0 3E    >39            BEQ   :2
85CF: C9 2C    >40            CMP   #´,´       Character must be either ´,´
85D1: D0 67    >41            BNE   GSNERR3     or ´-´
85D3: A6 C0    >42            LDX   LETINF     Process current letter
85D5: 20 34 86 >43            JSR   RDEFSUB
85D8: 10 E6    >44            BPL   ]LOOP      Always
85DA: 20 C4 7A >45   :2       JSR   RST100     Range:get the upper range let.
85DD: 20 D4 82 >46            JSR   MISLETC
85E0: C5 C0    >47            CMP   LETINF     Must not < 1st letter
85E2: 90 56    >48            BCC   GSNERR3
85E4: AA       >49            TAX              ;Into X for processing
85E5: 20 34 86 >50   ]JLOOP   JSR   RDEFSUB    process current letter within
85E8: CA       >51            DEX
85E9: E4 C0    >52            CPX   LETINF     Loop until 1st letter
85EB: B0 F8    >53            BCS   ]JLOOP
```

```
85ED: 20 29 86 >54            JSR   :LBS00
85F0: C9 2C    >55            CMP   #´,´
85F2: D0 46    >56            BNE   GSNERR3
85F4: F0 CA    >57            BEQ   ]LOOP       Always
85F6: 20 C4 7A >58   :LBS00   JSR   RST100
85F9: D0 3E    >59            BNE   R           Do not return if EOI
85FB: 68       >60            PLA
85FC: 68       >61            PLA
85FD: A6 C0    >62   :FIN     LDX   LETINF
85FF: F0 39    >63            BEQ   GSNERR3     Whaever args, process last letter
8601: A5 C1    >64   RDEFSUB  LDA   TYPMOD

Unknown label in line: 605 >65
               >65            STA   TYPLET-´A´,X
8605: 60       >66   R        RTS
8606: 4C C9 DE >67   GSNERR3  JMP   SYNERR
               >68
               >125
8609: 20 C4 7A >142  ROUT1Y   JSR   RST100
860C: 48       >143           PHA

Unknown label in line: 605 >144
               >144  ROUT1X   LDA   TVNORA,X
860F: 04 81    >145           TSB   VARNAM

Unknown label in line: 605 >146
               >146           LDA   TVN1ORA,X
8613: 04 82    >147           TSB   VARNAM+1
8615: 20 53 E0 >148           JSR   $E053       Attention, il faudra chg.
8618: 68       >149           PLA
8619: 60       >150           RTS
               >151
               >179
                606

Unknown label in line: 607
                607  XFRMMOT1 LDA   TYPLET-´A´,X
                608  XFROMMOT
                610  * X=0 for ´%´, 1 for ´$´ and 2 for ´!´, 3 for ´.´
861C: A2
Unknown label in line: 611
                611           LDX   #TITVAL-MOTIF-1

Unknown label in line: 615
                615  ]LOOP    CMP   MOTIF,X
8621: F0 3A     616           BEQ   :0
8623: CA        617           DEX
8624: 10 F9     618           BPL   ]LOOP
8626: 60        619  :0       RTS
                620
                621  * Decrement TXTPTR
8627: A5 B8     622  DECTPTR  LDA   TXTPTR
8629: D0 39     623           BNE   :0
862B: C6 B9     624           DEC   TXTPTR+1
862D: C6 B8     625  :0       DEC   TXTPTR
862F: 60        626           RTS
                627
```

```
                      628  * Subroutine to patch CHRGET/CHRGOT in page zero
8630: A9 4C           629  SETUPB   LDA   #$4C       JMP absolute
8632: 85 B1           630           STA   $B1
8634: 85 BA           631           STA   $BA
8636: A9 A6           632           LDA   #DEBUTGET
8638: 85 B2           632           STA   $B2
863A: A9 7A           632           LDA   #>DEBUTGET
863C: 85 B3           632           STA   $B2+1
863E: A9 ED           633           LDA   #DEBUTGOT
8640: 85 BB           633           STA   $BB
8642: A9 7A           633           LDA   #>DEBUTGOT
8644: 85 BC           633           STA   $BB+1
8646: 60              634           RTS
                      635
                      636  SETUPD   STID  BANCLD;$9D72
8647: A9 89           636           LDA   #BANCLD
8649: 8D 72 9D        636           STA   $9D72
864C: A9 86           636           LDA   #>BANCLD
864E: 8D 73 9D        636           STA   $9D72+1
8651: 60              637           RTS
                      638
                      639  * Subr. called upon a BASIC cold boot (FP DOS command)
8652: A2 FF           640  BANCLD   LDX   #$FF
8654: 86 76           641           STX   $76
8656: A2 FB           642           LDX   #$FB
8658: 9A              643           TXS
8659: A9 28           644           LDA   #$28
865B: A0 F1           645           LDY   #$F1
865D: 85 01           646           STA   1
865F: 84 02           647           STY   2
8661: 85 04           648           STA   4
8663: 84 05           649           STY   5
8665: 20 73 F2        650           JSR   $F273
8668: A9 4C           651           LDA   #$4C       JMP absolute
866A: 85 00           652           STA   0
866C: 85 03           653           STA   3
866E: 85 90           654           STA   $90
8670: 85 0A           655           STA   $A
8672: A9 99           656           LDA   #$99
8674: A0 E1           657           LDY   #$E1
8676: 85 0B           658           STA   $B
8678: 84 0C           659           STY   $C
867A: 20 67 86        660           JSR   SETUPB     Install CHRGET/CHRGOT patch in pa
ge zero
867D: 4C 5C F1        661           JMP   $F15C      End of initialization in ROM
                      662
                      663  * Do the DOS init
                      664  NOUVIN   STID  $E000;$9D72
8680: A9 00           664           LDA   #$E000
8682: 8D 72 9D        664           STA   $9D72
8685: A9 E0           664           LDA   #>$E000
8687: 8D 73 9D        664           STA   $9D72+1
868A: A9 4C           665           LDA   #$4C       JMP absolute
868C: 8D C8 A2        666           STA   $A2C8
868F: A9 0B           667           LDA   #$B
8691: 20 AA A2        668           JSR   $A2AA
8694: A9 20           669           LDA   #$20
```

```
8696: 8D C8 A2  670           STA   $A2C8
8699: A5 45     671           LDA   OPRND+1
869B: D0 3D     672           BNE   :4          No error during DoClose
869D: 20 7E 86  673           JSR   SETUPD     Reinstall Peersoft
86A0: 4C C8 A6  674           JMP   $A6C8       before exiting
86A3: A2 60     675  :4       LDX   #$60
86A5: 8E E7 A2  676           STX   $A2E7
86A8: 20 D2 A2  677           JSR   $A2D2      Copy file manager parmlist
86AB: A9 4C     678           LDA   #$4C       JMP absolute
86AD: 8D E7 A2  679           STA   $A2E7
86B0: AD 00 9D  680           LDA   DBUFP
86B3: 8D 02 87  681           STA   E06+1
86B6: AD 01 9D  682           LDA   DBUFP+1
86B9: 8D 07 87  683           STA   E06+6
86BC: A9 D3     684           LDA   #$9CD3
86BE: 8D 00 9D  684           STA   DBUFP
86C1: A9 9C     684           LDA   #>$9CD3
86C3: 8D 01 9D  684           STA   DBUFP+1
86C6: 20 06 AB  685           JSR   $AB06      File manager main entry (INIT)
86C9: 08        686           PHP              ;Save status
                687  E06      STID  0;DBUFP    Reinstall Peersoft DOS features
86CA: A9 00     687           LDA   #0
86CC: 8D 00 9D  687           STA   DBUFP
86CF: A9 00     687           LDA   #>0
86D1: 8D 01 9D  687           STA   DBUFP+1
86D4: 20 7E 86  688           JSR   SETUPD
86D7: 28        689           PLP
86D8: 20 EB A6  690           JSR   $A6EB      process possible error after FM c
all
86DB: 4C 97 A3  691           JMP   $A397      Goto SAVE (HELLO) command handler
                692
                693  * RECON is a subroutine which scans BASIC program area
                694  * or input buffer for a Peersoft new keyword
                695  * 2 entry points:
                696  * RECON1 (BASIC statement execution): the pointer is TXTPTR
                697  * RECON (BASIC statement listing): the pointer is in A,Y
                698  * X value of 0: search for every new keyword (LIST)
                699  *            1: search only DEF patterns
                700  *            2: search only function statements
                701  *               (IIF, MOUSE and TIMER)
                702  *            3: search only MOUSE and TIMER keywords
                703  * On exit, Z bit set means no keyword found
                704  *                clear means keyword (index in IDMOCL)
86DE: A5 B8     705  RECON1   LDA   TXTPTR
86E0: A4 B9     706           LDY   TXTPTR+1
86E2: 85 06     707  RECON    STA   AUXPTR
86E4: 84 07     708           STY   AUXPTR+1

Unknown label in line: 709
                709  RECON2   LDA   TIDMOCL,X
86E8: 85 BD     710           STA   IDMOCL

Unknown label in line: 711
                711           LDA   TOFFIN,X

Unknown label in line: 712
                712           STA   IFDEF
```

```
Unknown label in line: 713
                713           LDA   TOFFIN2,X

Unknown label in line: 714
                714           STA   IFIIF
86F2: E6 BD     715  :1       INC   IDMOCL
86F4: A4 BD     716           LDY   IDMOCL

Unknown label in line: 717
                717           LDX   TOFFST,Y
86F8: 86 C2     718           STX   OFFSET
86FA: A0 00     719           LDY   #0

Unknown label in line: 720
                720  ]LOOP    LDA   TMOCL,X
86FE: F0 4A     721           BEQ   :4          Keyword found: exit
8700: C9 FF     722           CMP   #$FF        End of table?
8702: F0 46     723           BEQ   :4          Yes: no keyword found
8704: D1 06     724           CMP   (AUXPTR),Y Current character match?
8706: D0 26     725           BNE   :1          no: try next keyword from table
8708: E8        726           INX               ;Next char. from current keyword
8709: C8        727           INY
870A: D0 F0     728           BNE   ]LOOP
                729
                730  :4       DO    KOPT-K65C02
870C: 1A        734           INC
870D: 60        736  RETURN   RTS
                737
                738           PUT   PEERLIST,D1
870E: 90 48    >1    STDLIS   BCC   STRTRNG
               >2
8710: F0 46    >3             BEQ   STRTRNG
8712: C9 C9    >4             CMP   #TOKMINUS
8714: F0 42    >5             BEQ   STRTRNG
8716: C9 2C    >6             CMP   #´,´
8718: D0 31    >7             BNE   RETURN
               >8

Unknown label in line: 738 >9
               >9    STRTRNG  JSR   DECOMPILE
871C: 20 0C DA >10            JSR   LINGET
871F: 20 1A D6 >11            JSR   FNDLIN
8722: 20 CC 7A >12            JSR   RST102
8725: F0 4F    >13            BEQ   MAINLIST
8727: C9 C9    >14            CMP   #TOKMINUS
8729: F0 43    >15            BEQ   ENDRNG
872B: C9 2C    >16            CMP   #´,´
872D: D0 1C    >17            BNE   RETURN
               >18
872F: 20 C4 7A >19   ENDRNG   JSR   RST100
8732: 20 0C DA >20            JSR   LINGET
8735: D0 14    >21            BNE   RETURN
               >22
8737: 68       >23   MAINLIST PLA
8738: 68       >24            PLA
8739: A5 50    >25            LDA   LINNUM      In case no second line given,
```

```
873B: 05 51    >26             ORA   LINNUM+1    let it be 65535
873D: D0 43    >27             BNE   NXLST
873F: C6 50    >28             DEC   LINNUM
8741: C6 51    >29             DEC   LINNUM+1
               >30
8743: A0 01    >31   NXLST     LDY   #1
8745: B1 9B    >32             LDA   (LOWTR),Y

Bad branch in line: 738 >33
               >33             BEQ   LISTED      End of program found
8749: 20 58 D8 >34             JSR   ISCNTC      Check for Ctrl-C keystroke
874C: 20 FB DA >35             JSR   CRDO
874F: C8       >36             INY
8750: B1 9B    >37             LDA   (LOWTR),Y   Line number in X,A
8752: AA       >38             TAX
8753: C8       >39             INY
8754: B1 9B    >40             LDA   (LOWTR),Y
8756: C5 51    >41             CMP   LINNUM+1    Beyond last line number?
8758: D0 43    >42             BNE   LSTD?
875A: E4 50    >43             CPX   LINNUM
875C: F0 41    >44             BEQ   LST1LIN

Bad branch in line: 738 >45
               >45   LSTD?     BCS   LISTED      Yes
               >46
8760: 84 85    >47   LST1LIN   STY   $85
8762: 64 BE    >55             STZ   MODREM
8764: 64 BF    >56             STZ   MODDAT
8766: 64 C0    >57             STZ   GFLAG
8768: 64 C1    >58             STZ   DEFFLG
876A: 20 F9 87 >60             JSR   VLINPRT     Print line #
876D: A9 20    >61   ]JLOOP    LDA   #32         Print space after line number
876F: A4 85    >62             LDY   $85
8771: 2C       >63             HEX   2C
8772: A9 2D    >64   L088      LDA   #´-´
8774: C9 22    >65   L08       CMP   #´"´        Is it ´"´?
8776: D0 47    >66             BNE   :9
8778: A5 C0    >67             LDA   GFLAG
877A: 49 FF    >68             EOR   #$FF
877C: 85 C0    >69             STA   GFLAG
877E: A9 22    >70             LDA   #´"´
               >71   * Now we test for an EOI
8780: 24 BE    >72   :9        BIT   MODREM      If a REM has been scanned in this
 line
8782: 30 4B    >73             BMI   SENDCHR
8784: 24 C0    >74             BIT   GFLAG       Are we within a string litteral?
8786: 30 47    >75             BMI   SENDCHR     Same output as for a REM
8788: C9 3A    >76             CMP   #´:´        Current char is EOI?
878A: D0 43    >77             BNE   SENDCHR
878C: 85 BF    >78             STA   MODDAT      MODDAT b7 forced to zero
878E: 85 C1    >79             STA   DEFFLG      DEFFLG b7 forced to zero
8790: 20 5C DB >80   SENDCHR   JSR   OUTDO       Print current char
8793: A5 24    >81             LDA   CH
8795: C9 21    >82             CMP   #33         Have we reached "right" edge of s
creen?
8797: 90 46    >83             BCC   NCR         No
8799: 20 FB DA >84             JSR   CRDO        Yes: print CR for next line
```

```
879C: A9 05    >85            LDA   #5
879E: 85 24    >86            STA   CH
               >87   * Next character from line
87A0: C8       >88   NCR      INY
87A1: B1 9B    >89            LDA   (LOWTR),Y
87A3: D0 57    >90            BNE   TOKEN?     Not end of line
87A5: 85 C1    >91            STA   DEFFLG
87A7: B2 9B    >98            LDA   (LOWTR)    Update next line pointer
87A9: AA       >99            TAX
87AA: A0 01    >100           LDY   #1
87AC: B1 9B    >102           LDA   (LOWTR),Y
87AE: 86 9B    >103           STX   LOWTR
87B0: 85 9C    >104           STA   LOWTR+1
87B2: D0 CE    >105           BNE   NXLST      Branch if not at program´s end
               >106
87B4: 20 FB DA >107  LISTED   JSR   CRDO
87B7: 4C D2 D7 >108           JMP   NEWSTT
87BA: 6C FA D6 >109  VLINPRT  JMP   ($D6FA)
87BD: AA       >110  TOKEN?   TAX               ;Character in X
87BE: A5 BE    >111           LDA   MODREM     Is litteral mode active?
87C0: 05 BF    >112           ORA   MODDAT
87C2: 05 C0    >113           ORA   GFLAG
87C4: 0A       >114           ASL
87C5: 8A       >115           TXA
87C6: B0 EB    >116           BCS   L08        Yes
87C8: 84 B5    >117           STY   YSAV
87CA: 98       >118           TYA               ;Compute Y, A = LOWTR + Y
87CB: A4 9C    >119           LDY   LOWTR+1
87CD: 65 9B    >120           ADC   LOWTR      Carry already clear
87CF: 90 40    >121           BCC   :14
87D1: C8       >122           INY
87D2: A2 00    >123  :14      LDX   #0
87D4: 20 19 87 >124           JSR   RECON      New BASIC keyword?
87D7: D0 72    >125           BNE   :23        Yes
               >126
87D9: A4 B5    >127           LDY   YSAV       Y = offset within line
87DB: B1 9B    >128           LDA   (LOWTR),Y  Current character
87DD: 10 D4    >129           BPL   L08        Not a token
87DF: 24 C1    >130           BIT   DEFFLG
87E1: 10 43    >131           BPL   :18
87E3: C9 C9    >132           CMP   #TOKMINUS
87E5: F0 CA    >133           BEQ   L088
87E7: C9 B2    >134  :18      CMP   #TOKREM    REM token?
87E9: D0 41    >135           BNE   :15
87EB: 66 BE    >136           ROR   MODREM     bit 7 to 1 in MODREM
87ED: C9 83    >137  :15      CMP   #TOKDATA   DATA token?
87EF: D0 41    >138           BNE   :16
87F1: 66 BF    >139           ROR   MODDAT     bit 7 to 1 in MODDAT
87F3: 48       >140  :16      PHA
87F4: 20 57 DB >141           JSR   OUTSPC
87F7: 68       >142           PLA
87F8: 48       >143           PHA
87F9: 20 99 88 >144           JSR   LTOKEN     Print Applesoft token
87FC: 68       >145           PLA
87FD: C9 D5    >146           CMP   #TOKUSR
87FF: 20 89 88 >147           JSR   COMLISO
8802: B0 44    >148           BCS   :17
```

```
8804: 84 85    >149           STY   $85
8806: 20 5C DB >150           JSR   OUTDO
8809: 4C 6D 87 >151  :17      JMP   ]JLOOP
               >152  * LIST a new BASIC statement
880C: 88       >153  :23      DEY
880D: A5 BD    >154           LDA   IDMOCL
880F: C9
Unknown label in line: 738 >155
               >155           CMP   #OFFDEF-TOFFST
8812: 90 41    >156           BCC   :39
8814: 66 C1    >157           ROR   DEFFLG
8816: 18       >158           CLC
8817: 98       >159  :39      TYA
8818: 65 B5    >160           ADC   YSAV
881A: 85 B5    >161           STA   YSAV
881C: 20 57 DB >162           JSR   OUTSPC
881F: A6 C2    >163           LDX   OFFSET     Get offset from new keyword table

Unknown label in line: 738 >164
               >164  ]LOOP    LDA   TMOCL,X
8823: F0 50    >165           BEQ   :29        End of keyword
8825: 30 44    >166           BMI   :27        Applesoft token: print it
8827: 20 5C DB >167           JSR   OUTDO      Normal text to output
882A: D0 46    >168           BNE   :28        Always
882C: 86 B4    >169  :27      STX   XSAV       Save offset
882E: 20 99 88 >170           JSR   LTOKEN     Print Applesoft token
8831: A6 B4    >171           LDX   XSAV
8833: E8       >172  :28      INX
8834: D0 EB    >173           BNE   ]LOOP      Always
8836: A5 BD    >174  :29      LDA   IDMOCL
8838: C9
Unknown label in line: 738 >175
               >175           CMP   #OFFUSR-TOFFST
883B: 20 89 88 >176           JSR   COMLISO
883E: B0 41    >177           BCS   :30
8840: 20 5C DB >178           JSR   OUTDO
8843: 20 57 DB >179  :30      JSR   OUTSPC
8846: A4 B5    >180  :31      LDY   YSAV
8848: 4C DF 87 >181           JMP   NCR
               >182
884B: 38       >183  COMLISO  SEC
884C: D0 4A    >184           BNE   :0
884E: A4 B5    >185           LDY   YSAV
8850: C8       >186           INY
8851: B1 9B    >187           LDA   (LOWTR),Y
8853: 20 D4 7A >188           JSR   COMRSTC
8856: B0 40    >189           BCS   :0
8858: 84 B5    >190           STY   YSAV
885A: 60       >191  :0       RTS
               >192
               >193  * Print Applesoft token
885B: 38       >194  LTOKEN   SEC
885C: E9 7F    >195           SBC   #$7F
885E: AA       >196           TAX              ;Index in X reg
885F: 84 85    >197           STY   $85
8861: A0 D0    >198           LDY   #TOKTABL-256
8863: 84 9D    >199           STY   FAC
```

```
                >200  * Line below is a substitute for LDY #>TOKTABL-256
8865: 88        >201            DEY
8866: 84 9E     >202            STY   FAC+1
8868: A0 FF     >203            LDY   #$FF
886A: CA        >204  :1        DEX
886B: F0 45     >205            BEQ   :3
886D: 20 2C D7  >206  ]LOOP     JSR   $D72C
8870: 10 FB     >207            BPL   ]LOOP
8872: 30 34     >208            BMI   :1
8874: 20 2C D7  >209  :3        JSR   $D72C
8877: 30 43     >210            BMI   :4
8879: 20 5C DB  >211            JSR   OUTDO
887C: D0 34     >212            BNE   :3
887E: A4 85     >213  :4        LDY   $85
8880: 4C 5C DB  >214            JMP   OUTDO
                 739
8883: D0 45      740  RRETURN   BNE   :0
8885: A9 FF      741            LDA   #$FF
8887: 85 86      742            STA   FORPNT+1
8889: 4C 71 D9   743            JMP   $D971
888C: 60         744  :0        RTS
                 745
888D: A9 AB      746  RONERR    LDA   #TOKGOTO
888F: 20 DA 82   747            JSR   NSYNCHR
8892: A5 B8      748            LDA   TXTPTR
8894: 85 F4      749            STA   TXTPSV
8896: A5 B9      750            LDA   TXTPTR+1
8898: 85 F5      751            STA   TXTPSV+1
889A: 38         752            SEC
889B: 66 D8      753            ROR   ERRFLG
889D: A5 75      754            LDA   CURLIN
889F: 85 F6      755            STA   CURLSV
88A1: A5 76      756            LDA   CURLIN+1
88A3: 85 F7      757            STA   CURLSV+1
88A5: 4C 95 D9   758            JMP   DATA
                 759
                 760  * New FRMEVL processing
                 761            PUT   PEERAROMBA,D2
                >1    TOKDIM    =     $86
                >2    TOKFRE    =     $D6
                >3    NEWGARBG  EQU   $E484
                >4    FREFAC    EQU   $E600
                >5    ENDCHR    EQU   $0E
                >6    STRNG1    EQU   $AC
                >7    VPNT      EQU   $A0
                >8    * When used in USR functions w 2 args, holdsin n
                >9    * the first arg expression type
                >10   GIVAYF    EQU   $E2F2
                >11   SNGFLT    EQU   $E301
                >12   MOVMF     EQU   $EB2B
                >13   LEVELPAR  EQU   IDMOCL
                >14
88A8: 20 C4 7A  >85   RDIM      JSR   RST100
88AB: 20 F5 8A  >86             JSR   NCHKOPN
88AE: 20 8A 7E  >87             JSR   NGETARPT
88B1: A0 04     >88             LDY   #4
88B3: B1 9B     >89             LDA   (LOWTR),Y
```

```
88B5: 29 0F    >90            AND   #$0F
88B7: 48       >91            PHA
88B8: B2 B8    >93            LDA   (TXTPTR)
88BA: C9 2C    >98            CMP   #´,´
88BC: D0 63    >99            BNE   :1
88BE: D4 9B    >101           PEI   LOWTR
88C0: 20 C4 7A >108           JSR   RST100
88C3: 20 1C 8B >109           JSR   NGETBYT    Index of dimension in X&FACLO
88C6: 8A       >110           TXA
88C7: F0 62    >111           BEQ   GOIQ
88C9: 68       >112           PLA
88CA: 85 9B    >113           STA   LOWTR
88CC: 68       >114           PLA
88CD: 85 9C    >115           STA   LOWTR+1
88CF: 68       >116           PLA
88D0: 38       >117           SEC
88D1: E5 A1    >118           SBC   FACLO
88D3: 90 56    >119           BCC   GOIQ
88D5: 0A       >120           ASL              ;Incidently clears the carry
88D6: 69 05    >121           ADC   #5         Because of carry clear
88D8: A8       >122           TAY
88D9: B1 9B    >123           LDA   (LOWTR),Y
88DB: AA       >124           TAX
88DC: C8       >125           INY
88DD: B1 9B    >126           LDA   (LOWTR),Y
88DF: A8       >127           TAY
88E0: 8A       >128           TXA
88E1: 90 42    >129           BCC   :0         Always
               >130  :1       MPLY
88E3: 7A       >130           PLY
88E4: A9 00    >132           LDA   #0
88E6: 38       >136           SEC
88E7: 20 F2 E2 >137  :0       JSR   GIVAYF
88EA: 4C EF 8A >138           JMP   NCHKCLS
               >139
88ED: 4C 99 E1 >140  GOIQ     JMP   GOIQERR    Raise a ILLEGAL QUANTITY ERROR
               >141
88F0: 20 FA 8A >142  RVRAI    JSR   NFRMEVL    True: evaluate second argument
88F3: 20 F2 8A >143           JSR   NCHKCOM    Skip the comma and 3rd expr.
88F6: A9 29    >144           LDA   #´)´       until end of function detected
               >145
               >146  * This subroutine will skip program text until an
               >147  * end character is scanned.
88F8: 85 0E    >148  SKIPC    STA   ENDCHR
88FA: A0 00    >149           LDY   #0
88FC: 84 BD    >150           STY   LEVELPAR   Parenthesis level
88FE: 84 C0    >151           STY   GFLAG      String litteral parsing flag
8900: 88       >152           DEY
8901: C8       >153  ]LOOP    INY
8902: B1 B8    >154           LDA   (TXTPTR),Y
8904: F0 74    >155           BEQ   LGSYNERR
8906: C9 22    >156           CMP   #´"´
8908: D0 46    >157           BNE   :0
890A: A5 C0    >158           LDA   GFLAG      Inverse GFLAG b7
890C: 49 80    >159           EOR   #$80
890E: 85 C0    >160           STA   GFLAG
8910: B0 EF    >161           BCS   ]LOOP      Always
```

```
8912: 24 C0    >162  :0       BIT   GFLAG      Within litteral string
8914: 30 EB    >163          BMI   ]LOOP       so loop for next character.
8916: C9 3A    >164          CMP   #´:´        End of instruction?
8918: F0 60    >165          BEQ   LGSYNERR    SYNTAX ERROR if so
891A: C9 28    >166          CMP   #´(´
891C: D0 42    >167          BNE   :1
891E: E6 BD    >168          INC   LEVELPAR
8920: B0 DF    >169          BCS   ]LOOP       Always
8922: C9 29    >170  :1       CMP   #´)´
8924: D0 46    >171          BNE   :2
8926: A6 BD    >172          LDX   LEVELPAR
8928: F0 46    >173          BEQ   :3
892A: C6 BD    >174          DEC   LEVELPAR
892C: 10 D3    >175          BPL   ]LOOP
892E: A6 BD    >176  :2       LDX   LEVELPAR
8930: D0 CF    >177          BNE   ]LOOP
8932: C5 0E    >178  :3       CMP   ENDCHR
8934: D0 CB    >179          BNE   ]LOOP
8936: 20 98 D9 >180          JSR   ADDON       Add Y to TXTPTR
8939: 4C C4 7A >181          JMP   RST100
               >182
893C: 4C C9 DE >183  LGSYNERR JMP   SYNERR     Vector to SYNTAX ERROR
               >184
               >185  * Handles the IIF function
893F: 20 F2 8A >186  RIIF     JSR   NCHKCOM    Check for trailing comma
8942: A6 9D    >187          LDX   FAC         True or false value?
8944: D0 E8    >188          BNE   RVRAI       True: then skip second arg.
8946: A9 2C    >189          LDA   #´,´
8948: 20 36 89 >190          JSR   SKIPC       Skip 2nd expression
               >191  * Evaluate 3rd arg. and check for closing parenthesis
894B: 4C EC 8A >192          JMP   NPARCHK+3
               >193
894E: 20 FA 8A >194  NFRMNUM  JSR   NFRMEVL    Get scalar valueH
8951: 4C 6A DD >195          JMP   CHKNUM     Ensure numeric value
               >196
8954: 4C F9 EA >197  ]LOOP    JMP   MOVFM
8957: D4 A0    >200          PEI   VPNT
8959: 20 E9 DE >201          JSR   $DEE9
895C: 68       >213          PLA
895D: 20 11 8B >214          JSR   LBS81
8960: 4C 0F 8A >215          JMP   XSUITE
               >216
               >217  * Takes care of the ´@´ processing
               >218  * Refactor part of the FRMEVL ROM routine
8963: 20 C4 7A >219  FRMELMLP JSR   RST100
8966: B0 45    >220  FRMELM   BCS   :2          Branch iif not a digit
8968: 64 C7    >228  :1       STZ   INTTYPSV
896A: 64 C8    >229          STZ   VALTYPSV
896C: 4C 4A EC >230          JMP   $EC4A
896F: C9 2E    >232  :2       CMP   #´.´
8971: F0 33    >233          BEQ   :1
8973: 20 7D E0 >234          JSR   ISLETC

Bad branch in line: 761 >235
               >235          BCC   L3
8978: AA       >236          TAX
8979: 30 66    >237          BMI   :77
```

```
897B: C9 49    >238           CMP   #´I´
897D: F0 46    >239           BEQ   :80
897F: C9 4D    >240           CMP   #´M´
8981: F0 42    >241           BEQ   :80
8983: C9 54    >242           CMP   #´T´
8985: D0 5A    >243           BNE   :77
               >244  * Might be the IIF() function
8987: A2 02    >245  :80       LDX   #2
8989: 20 15 87 >246           JSR   RECON1
898C: F0 53    >247           BEQ   :77
898E: 20 98 D9 >248           JSR   ADDON
8991: A5 BD    >249           LDA   IDMOCL
8993: 48       >250           PHA
8994: 20 F5 8A >251           JSR   NCHKOPN
8997: 20 8C 89 >252           JSR   NFRMNUM    Get operand numeric value
899A: 68       >253           PLA              ;Recall IDMOCL from stack
899B: 38       >254           SEC
899C: E9
Unknown label in line: 761 >255
               >255           SBC   #OFFMOU-TOFFST
899F: 90 DC    >256           BCC   RIIF
               >257  * Space for MOUSE and TIMER functions
               >258  * ...: to be continued
89A1: 4C D6 90 >259           JMP   MTFUNC
               >260  * Alphabetic character: variable name
89A4: A2 00    >261  :77       LDX   #0
89A6: 86 10    >262           STX   DIMFLG
89A8: B2 B8    >266           LDA   (TXTPTR)
89AA: 20 96 7E >268           JSR   NPTRGET1
               >269  RFFVL     EQU   *-1
89AD: 85 A0    >270           STA   VPNT
89AF: 84 A1    >271           STY   VPNT+1
89B1: A6 11    >272           LDX   VALTYP
89B3: F0 41    >273           BEQ   :41
89B5: 64 AD    >279           STZ   STRNG1+1
89B7: D0 56    >280           BNE   XSUITE     Always
89B9: A6 12    >282  :41       LDX   INTTYP
89BB: 10 97    >283           BPL   ]LOOP
89BD: E0 81    >284           CPX   #$81

Unknown label in line: 761 >285
               >285           BNE   H16B       Branch if int16bit variable
89C1: A2 00    >286           LDX   #0
89C3: B2 83    >288           LDA   (VARPNT)
89C5: 10 06    >292           BPL   *+8

Unknown label in line: 761 >293
               >293           BIT   WMODE
89C9: 30 01    >294           BMI   *+3
89CB: CA       >295           DEX              ;Poids fort dans X
89CC: A8       >296           TAY              ;Poids faible dans Y
89CD: 8A       >297           TXA              ;Poids fort dans A
89CE: 20 F2 E2 >298           JSR   GIVAYF     Convert A, Y to FP
89D1: A5 11    >299  XSUITE    LDA   VALTYP
89D3: 85 C8    >300  RET3      STA   VALTYPSV
89D5: 60       >301  ]RET      RTS
               >302
```

```
89D6: C9 C8    >303  L3       CMP   #TOKADD    Unary + operator: loop
89D8: F0 C7    >304           BEQ   FRMELMLP
89DA: C9 22    >305           CMP   #´"´
89DC: D0 48    >306           BNE   :4
89DE: 20 81 DE >307           JSR   $DE81
89E1: A9 FF    >308           LDA   #$FF
89E3: 30 2C    >309           BMI   RET3       Always
89E5: 4C C6 83 >310  ]LOOP    JMP   RUSR
89E8: C9 D5    >311  :4       CMP   #TOKUSR
89EA: F0 F9    >312           BEQ   ]LOOP
89EC: A2
Unknown label in line: 761 >313
               >313           LDX   #TOKMTIFE-TOKMOTIF-1

Unknown label in line: 761 >314
               >314  ]LOOP    CMP   TOKMOTIF,X
89F1: D0 46    >315           BNE   :NOK
89F3: A8       >325           TAY
89F4: 8A       >326           TXA
89F5: 0A       >327           ASL
89F6: AA       >328           TAX
89F7: 98       >329           TYA

Unknown label in line: 761 >330
               >330           JMP   (TOKMPF,X)
89FA: CA       >332  :NOK     DEX
89FB: 10 F2    >333           BPL   ]LOOP
89FD: C9 40    >334  :6       CMP   #´@´
89FF: D0 4F    >335           BNE   :78
8A01: A5 C8    >336           LDA   VALTYPSV
8A03: 85 11    >337           STA   VALTYP
8A05: 30 43    >338           BMI   :60
8A07: A5 C7    >339           LDA   INTTYPSV
8A09: 85 12    >340           STA   INTTYP
8A0B: 4C C4 7A >341  :60      JMP   RST100
8A0E: 4C E6 88 >342  :79      JMP   RDIM
8A11: C9 86    >343  :78      CMP   #TOKDIM
8A13: F0 38    >344           BEQ   :79
               >345
8A15: C9 D2    >346  :7       CMP   #TOKSGN
8A17: B0 57    >347           BCS   :10
8A19: C9 23    >348           CMP   #´#´
8A1B: F0 03    >349           BEQ   *+5
8A1D: 4C E9 8A >350           JMP   NPARCHK
               >351  * Handle the ´#´ pattern in a FOREACH loop

Unknown label in line: 761 >352
               >352           LDY   AEI

Unknown label in line: 761 >353
               >353           LDA   AEI+1
8A24: EB       >355           XBA
8A25: 20 F2 E2 >359           JSR   GIVAYF
8A28: EB       >361           XBA
8A29: 20 14 8B >365           JSR   LBS80
8A2C: 4C C4 7A >366           JMP   RST100
8A2F: 0A       >367  :10      ASL
```

```
8A30: 48       >368           PHA
8A31: AA       >369           TAX
8A32: 20 C4 7A >370           JSR   RST100
8A35: E0 CF    >371           CPX   #$CF
8A37: 90 53    >372           BCC   :11
8A39: 20 F5 8A >373           JSR   NCHKOPN
8A3C: 20 FA 8A >374           JSR   NFRMEVL
8A3F: 20 F2 8A >375           JSR   NCHKCOM
8A42: 20 6C DD >376           JSR   CHKSTR
8A45: FA       >377           PLX
8A46: 20 D3 8A >378           JSR   COMCMPLX
8A49: 80 50    >382           BRA   :14
8A4B: 20 E9 8A >384  :11      JSR   NPARCHK
8A4E: 7A       >385           PLY
8A4F: C0 C8    >386           CPY   #TOKSTRD+TOKSTRD
8A51: F0 45    >387           BEQ   :15
8A53: C0 CE    >388           CPY   #TOKCHRD+TOKCHRD
8A55: D0 72    >389           BNE   :13
8A57: 20 DB 8A >390  :15      JSR   CALLFUNC
8A5A: A9 FF    >391  :14      LDA   #$FF
8A5C: 85 C8    >392           STA   VALTYPSV
8A5E: 60       >393  ]RET     RTS
8A5F: A5 11    >394  ]LOOP    LDA   VALTYP
8A61: D0 5D    >395           BNE   :19
8A63: 18       >396           CLC
8A64: 20 D6 7C >397           JSR   NROUT
8A67: A2 00    >398           LDX   #0
8A69: A5 A0    >399           LDA   FAC+3
8A6B: D0 56    >400           BNE   :2
8A6D: A5 A1    >401           LDA   FAC+4
8A6F: C9 01    >402           CMP   #1
8A71: D0 50    >403           BNE   :2
8A73: A2 03    >404           LDX   #3
8A75: 20 2F 82 >405           JSR   ZRTAUX
8A78: A5 AE    >406           LDA   STRNG2+1
8A7A: A4 AD    >407           LDY   STRNG2
8A7C: 4C 0C 8B >408           JMP   NWGVAYF
8A7F: 20 00 E6 >409  :19      JSR   FREFAC
8A82: 20 84 E4 >410  :2       JSR   NEWGARBG
8A85: 4C 02 8B >411           JMP   HE2E8
               >412
8A88: C0 AC    >413  :13      CPY   #TOKFRE+TOKFRE
8A8A: F0 D3    >414           BEQ   ]LOOP
8A8C: 20 DB 8A >415           JSR   CALLFUNC
8A8F: 4C 6A DD >416           JMP   CHKNUM
               >417
               >418  COMCMPLX DO    KOPT16
8A92: D4 A0    >419           PEI   FACMO
8A94: DA       >426           PHX
8A95: 20 1C 8B >427           JSR   NGETBYT
8A98: 7A       >428           PLY
8A99: DA       >429           PHX
               >430
8A9A: B9 DC CF >431  CALLFUNC LDA   $CFDC,Y
8A9D: 85 91    >432           STA   $91
8A9F: B9 DD CF >433           LDA   $CFDD,Y
8AA2: 85 92    >434           STA   $92
```

```
8AA4: 20 90 00 >435           JSR   $90
8AA7: 60       >436           RTS
               >437
8AA8: 20 F5 8A >438  NPARCHK  JSR   NCHKOPN
8AAB: 20 FA 8A >439           JSR   NFRMEVL
               >440
8AAE: A9 29    >441  NCHKCLS  LDA   #´)´
8AB0: 2C       >442           HEX   2C
8AB1: A9 2C    >443  NCHKCOM  LDA   #´,´
8AB3: 2C       >444           HEX   2C
8AB4: A9 28    >445  NCHKOPN  LDA   #´(´
8AB6: 4C DA 82 >446           JMP   NSYNCHR
               >447
8AB9: 20 7B DD >448  NFRMEVL  JSR   FRMEVL
8ABC: A5 11    >449           LDA   VALTYP
8ABE: 85 C8    >450           STA   VALTYPSV
8AC0: 60       >451  ]RET     RTS
               >452
8AC1: 38       >453  HE2E8    SEC
8AC2: A5 6F    >454           LDA   FRETOP
8AC4: E5 6D    >455           SBC   STREND
8AC6: A8       >456           TAY
8AC7: A5 70    >457           LDA   FRETOP+1
8AC9: E5 6E    >458           SBC   STREND+1
8ACB: 48       >459  NWGVAYF  PHA
8ACC: 20 F2 E2 >460           JSR   GIVAYF
8ACF: 68       >461           PLA

Unknown label in line: 761 >462
               >462  LBS81    AND   WMODE
8AD2: 10 EC    >463  LBS80    BPL   ]RET
8AD4: 20 C9 92 >464           JSR   GP65536
8AD7: 4C BE E7 >465           JMP   FADD
               >466
8ADA: 20 F8 E6 >467  NGETBYT  JSR   GETBYT
8ADD: 48       >468           PHA
8ADE: 20 60 7C >469           JSR   SETITS
8AE1: 64 C8    >474           STZ   VALTYPSV
8AE3: 68       >476           PLA
8AE4: 60       >477  MFIN     RTS
                762
8AE5: 20 4C E7  763  ROUT11   JSR   COMBYTE    Get VTAB value in X
8AE8: 20 59 F2  764           JSR   $F259      Do the VTAB
8AEB: 20 4C E7  765           JSR   COMBYTE
8AEE: 20 EA F7  766           JSR   $F7EA      Do the HTAB
8AF1: 20 CC 7A  767           JSR   RST102
8AF4: F0 55     768           BEQ   :0
8AF6: 20 F2 8A  769           JSR   NCHKCOM
8AF9: A5 F1     770           LDA   $F1        Save current SPEED
8AFB: 48        771           PHA
8AFC: A9 01     772           LDA   #1         Fastest speed..
8AFE: 85 F1     773           STA   $F1
8B00: 20 CC 7A  774           JSR   RST102
8B03: 20 D5 DA  775           JSR   $DAD5      Do the PRINT
8B06: 68        776           PLA              ;restore original SPEED
8B07: 85 F1     777           STA   $F1
8B09: 60        778  :0       RTS
```

```
                        779
8B0A: 20 F2 8A  780  ROUTGEN  JSR   NCHKCOM
8B0D: 20 1C 8B  781           JSR   NGETBYT
8B10: 8A        782           TXA
8B11: F0 61     783           BEQ   ROUT0
8B13: E0 0B     784           CPX   #11
8B15: F0 10     785           BEQ   ROUT11
8B17: E0 0A     786           CPX   #10
8B19: D0 45     787           BNE   :2
8B1B: 4C EC 8E  788           JMP   ROUT10
8B1E: E0 08     789  :2       CPX   #8
8B20: D0 45     790           BNE   :1

Unknown label in line: 791
                791           JMP   ROUT8
8B24: E0 05     792  :1       CPX   #5
8B26: D0 46     793           BNE   :0
8B28: 4C 49 8D  794           JMP   KILLEMAL
8B2B: B0 F9     795  :0       BCS   MFIN
8B2D: E0 04     796           CPX   #4

Bad branch in line: 797
                797           BEQ   ROUT4
8B31: A5 69     798  ROUT0    LDA   VARTAB
8B33: 85 06     799           STA   AUXPTR
8B35: A5 6A     800           LDA   VARTAB+1
8B37: 85 07     801           STA   AUXPTR+1
                802
8B39: 20 CC 7A  803  ]LOOP    JSR   RST102
8B3C: F0 E8     804           BEQ   MFIN
8B3E: 20 F2 8A  805           JSR   NCHKCOM
8B41: 20 E9 8C  806           JSR   NPTRGETX
8B44: A5 9B     807           LDA   LOWTR
8B46: C5 06     808           CMP   AUXPTR
8B48: A5 9C     809           LDA   LOWTR+1
8B4A: E5 07     810           SBC   AUXPTR+1
8B4C: 90 D8     811           BCC   MFIN
8B4E: A0 00     812           LDY   #0
8B50: B1 9B     813  ]JLOOP   LDA   (LOWTR),Y
8B52: AA        814           TAX
8B53: B1 06     815           LDA   (AUXPTR),Y
8B55: 91 9B     816           STA   (LOWTR),Y
8B57: 8A        817           TXA
8B58: 91 06     818           STA   (AUXPTR),Y
8B5A: C8        819           INY
8B5B: C0 07     820           CPY   #7
8B5D: 90 F1     821           BCC   ]JLOOP
8B5F: 18        822           CLC
8B60: 98        823           TYA
8B61: 65 06     824           ADC   AUXPTR
8B63: 85 06     825           STA   AUXPTR
8B65: 90 D2     826           BCC   ]LOOP
8B67: E6 07     827           INC   AUXPTR+1
8B69: B0 CE     828           BCS   ]LOOP      Always
                829
8B6B: 4C 76 DD  830  GGO2TMER JMP   GOTMIERR
                831
```

```
8B6E: A9 04     832  ROUT4    LDA   #4          Ensure enough room on stack
8B70: 20 D6 D3  833           JSR   CHKMEM      7 bytes so 4 16bit words
8B73: 68        834           PLA               ;Pull return adress
8B74: 68        835           PLA
8B75: 20 F2 8A  836           JSR   NCHKCOM
8B78: 20 8E 7E  837           JSR   NPTRGTX
8B7B: 24 12     838           BIT   INTTYP
8B7D: 10 2F     839           BPL   GGO2TMER
8B7F: A5 9B     840           LDA   LOWTR
8B81: C5 6B     841           CMP   ARYTAB

Unknown label in line: 842
                842           STA   ITVADDR
8B85: A5 9C     843           LDA   LOWTR+1

Unknown label in line: 844
                844           STA   ITVADDR+1
8B89: E5 6C     845           SBC   ARYTAB+1
8B8B: B0 21     846           BCS   GGO2TMER
8B8D: A5 F8     847           LDA   REMSTK

Unknown label in line: 848
                848           STA   SPROOT
                849  * Reinit the alive context markers
8B91: A9 FF     850           LDA   #$FF
8B93: A2
Unknown label in line: 851
                851           LDX   #TABOFT-TABOFB

Unknown label in line: 852
                852  ]LOOP    STA   TABOFT-1,X
8B98: CA        853           DEX
8B99: D0 FB     854           BNE   ]LOOP
8B9B: 86 C0     855           STX   IDX0        Starting index: 0
8B9D: 20 CC 7A  856  ]LOOP    JSR   RST102
8BA0: F0 55     857           BEQ   XMFIN       End of instruction
8BA2: 20 F2 8A  858           JSR   NCHKCOM
8BA5: 20 CE 92  859           JSR   NGTA2
8BA8: 90 76     860           BCC   XMFIN1
8BAA: 20 23 8C  861           JSR   LBS04
8BAD: E6 C0     862           INC   IDX0
8BAF: D0 EC     863           BNE   ]LOOP
                864
8BB1: A5 C0     865  XMFIN    LDA   IDX0
8BB3: F0 67     866           BEQ   :0
8BB5: A9 80     867           LDA   #$80

Unknown label in line: 868
                868           STA   MTACTV
8BB9: 20 BC 8D  869           JSR   SETLTR
8BBC: 20 20 8C  870           JSR   XMFIN1
8BBF: A9 00     878           LDA   #0
8BC1: 24 D8     879           BIT   ERRFLG
8BC3: 10 01     880           BPL   *+3
8BC5: 1A        881           INC
8BC6: A0 1A     883           LDY   #26
8BC8: 91 9B     884           STA   (LOWTR),Y
```

```
8BCA: 20 23 8E  885           JSR   SAVERC
8BCD: A2 00     886           LDX   #0

Unknown label in line: 887
                887           STX   INDX
8BD1: 4C 65 8D  888           JMP   RESTOR1
8BD4: 60        889  :0       RTS
                890
8BD5: 28        891  XMFIN2   PLP
8BD6: 68        892           PLA
8BD7: 68        893           PLA
8BD8: 4C 95 D9  894  XMFIN1   JMP   DATA
                895
                896  * Handle a single entry (index in IDX0)
                897  LBS04
                898  * Array base address in (LOWTR, LOWTR+1)
8BDB: A6 C0     899           LDX   IDX0
8BDD: A5 9B     900           LDA   LOWTR
8BDF: 85 06     901           STA   AUXPTR
8BE1: E5 6B     902           SBC   ARYTAB     C already set

Unknown label in line: 903
                903           STA   TABOFB,X
8BE5: 08        904           PHP
8BE6: A5 9C     905           LDA   LOWTR+1
8BE8: 85 07     906           STA   AUXPTR+1
                907  * Is local error handling desired
8BEA: 20 F2 8A  908           JSR   NCHKCOM
8BED: 20 F8 E6  909           JSR   GETBYT
                910  * Offset 24 for local error handling flag
8BF0: A0 1A     911           LDY   #26
8BF2: E0 02     912           CPX   #2
8BF4: D0 4F     913           BNE   :0
8BF6: CA        914           DEX
8BF7: 24 D8     915           BIT   ERRFLG
8BF9: 30 4A     916           BMI   :0
8BFB: CA        917           DEX
8BFC: 8A        918  :0       TXA
8BFD: 91 06     919           STA   (AUXPTR),Y
8BFF: F0 57     920           BEQ   :1
8C01: A0 19     921           LDY   #26-1

Unknown label in line: 922
                922  ]LOOP    LDX   P0OFFSET-8,Y
8C05: B5 00     923           LDA   0,X
8C07: 91 06     924           STA   (AUXPTR),Y
8C09: 88        925           DEY
8C0A: E0 F4     926           CPX   #TXTPSV
8C0C: D0 F5     927           BNE   ]LOOP
                928  * Offsets 27 and 28 for swapped in machine code routine
8C0E: A9 1C     929  :1       LDA   #28
8C10: 20 D2 8C  930           JSR   LBS041
                931  * Offsets 29 and 30 for swapped out machine code routine
8C13: A9 1E     932           LDA   #30
8C15: 20 D2 8C  933           JSR   LBS041
8C18: 20 F2 8A  934           JSR   NCHKCOM
8C1B: 20 0C DA  935           JSR   LINGET
```

```
8C1E: 20 1A D6  936           JSR   FNDLIN
8C21: 90 FA     937           BCC   XMFIN2     Non existent line: exit
                938  * Offsets 0 and 1 for array name
                939  * Offsets 2 and 3 for offset to next array
                940  * Offset 4 for number of dimension
                941  * Offsets 5 and 6 for last dimension value
8C23: A0 04     942           LDY   #4
8C25: B1 06     943           LDA   (AUXPTR),Y
8C27: 49 41     944           EOR   #%01000001 Must be 16bits integer and
8C29: D0 F2     945           BNE   XMFIN2     # of dimensions must be 1
8C2B: A5 07     946           LDA   AUXPTR+1
8C2D: 28        947           PLP                ;Restaure Carry from previous SBC
8C2E: E5 6C     948           SBC   ARYTAB+1
8C30: A6 C0     949           LDX   IDX0

Unknown label in line: 950
                950           STA   TABOFT,X
                951  * Offset 7 and 8 for storing SP value
                952  * Integer variable value storage order
8C34: A0 07     953           LDY   #7
8C36: A9 00     954           LDA   #0
8C38: 91 06     955           STA   (AUXPTR),Y
8C3A: C8        956           INY
8C3B: A5 F8     957           LDA   REMSTK
8C3D: E9 07     958           SBC   #7         ;Carry already set
8C3F: 91 06     959           STA   (AUXPTR),Y
8C41: C8        960           INY
                961  * Offset 9 and 10 for LINNUM storage
                962  * (natural storage order)
8C42: A5 50     963           LDA   LINNUM
8C44: 91 06     964           STA   (AUXPTR),Y
8C46: C8        965           INY
8C47: A5 51     966           LDA   LINNUM+1
8C49: 91 06     967           STA   (AUXPTR),Y
8C4B: C8        968           INY
                969  * Offset 11 and 12 for TXTPTR storage
                970  * (natural storage order)
8C4C: A5 9B     971           LDA   LOWTR
8C4E: 69 03     972           ADC   #4-1       Because Carry already set
8C50: 91 06     973           STA   (AUXPTR),Y
8C52: C8        974           INY
8C53: A5 9C     975           LDA   LOWTR+1
8C55: 69 00     976           ADC   #0
8C57: 91 06     977           STA   (AUXPTR),Y
8C59: C8        978           INY
                979  * Offset 13 and 14 for OLDTEXT storage
                980  * (natural storage order)
8C5A: A5 9B     981           LDA   LOWTR
8C5C: 69 04     982           ADC   #4
8C5E: 91 06     983           STA   (AUXPTR),Y
8C60: C8        984           INY
8C61: A5 9C     985           LDA   LOWTR+1
8C63: 69 00     986           ADC   #0
8C65: 91 06     987           STA   (AUXPTR),Y
8C67: A0 1F     988           LDY   #31
                989  * Offsset 31 and above for stack content storage
                990  * from current SP to SPROOT
```

```
                        991  * For the time being (init), prepare a GOSUB frame
8C69: A9 B0             992            LDA   #TOKGOSUB
8C6B: A2 03             993            LDX   #3
8C6D: 91 06             994  ]JLOOP    STA   (AUXPTR),Y Do not mind calling CURLIN
8C6F: C8                995            INY
8C70: CA                996            DEX
8C71: D0 FA             997            BNE   ]JLOOP
8C73: A5 79             998            LDA   OLDTPTR
8C75: 91 06             999            STA   (AUXPTR),Y
8C77: C8                1000           INY
8C78: A5 7A             1001           LDA   OLDTPTR+1
8C7A: 91 06             1002           STA   (AUXPTR),Y
8C7C: C8                1003           INY
8C7D: A9 D1             1004           LDA   #NEWSTT-1
8C7F: 91 06             1005           STA   (AUXPTR),Y
8C81: C8                1006           INY
8C82: A9 D7             1007           LDA   #>NEWSTT-1
8C84: 91 06             1008           STA   (AUXPTR),Y
8C86: 60                1009           RTS
                        1010
8C87: 48                1011 LBS041    PHA
8C88: 20 F2 8A          1012           JSR   NCHKCOM
8C8B: 20 67 DD          1013           JSR   FRMNUM
8C8E: 20 52 E7          1014           JSR   GETADR
8C91: 7A                1015           PLY
8C92: A5 51             1016           LDA   LINNUM+1
8C94: 91 06             1017           STA   (AUXPTR),Y
8C96: F0 50             1018           BEQ   :0
8C98: 88                1019           DEY
8C99: A5 50             1020           LDA   LINNUM
8C9B: 91 06             1021           STA   (AUXPTR),Y
8C9D: 60                1022 :0        RTS
                        1023
                        1024 NPTRGETX  DO    KOPT-K65C02
8C9E: 64 82             1028           STZ   VARNAM+1
8CA0: 20 D4 82          1030           JSR   MISLETC
8CA3: 85 81             1031           STA   VARNAM
8CA5: 20 C4 7A          1032           JSR   RST100
8CA8: 90 50             1033           BCC   :0
8CAA: 20 7D E0          1034           JSR   ISLETC
8CAD: 90 61             1035           BCC   :3
8CAF: 85 82             1036 :0        STA   VARNAM+1
8CB1: 20 C4 7A          1037 ]LOOP     JSR   RST100
8CB4: 90 FB             1038           BCC   ]LOOP
8CB6: 20 7D E0          1039           JSR   ISLETC
8CB9: B0 F6             1040           BCS   ]LOOP
8CBB: 90 53             1041           BCC   :3
8CBD: 20 5E 86          1042 :2        JSR   DECTPTR
8CC0: A6 81             1043           LDX   VARNAM

Unknown label in line: 1044
                        1044           LDA   TYPLET-´A´,X
8CC4: A2 03             1046 :3        LDX   #3
8CC6: 20 55 86          1050           JSR   XFROMMOT+2
8CC9: D0 3D             1051           BNE   :2
8CCB: 4C 3D 86          1052           JMP   ROUT1Y
                        1053
```

```
Unknown label in line: 1054
                1054 RNEWISUI BIT   MTACTV

Bad branch in line: 1055
                1055          BPL   RESTORD
                1056
                1057          PUT   PEERMTK
               >1    * Main Active MT entry point
8CD2: BA       >2    RMTCTRL  TSX              ;Test for an exhausted thread?

Unknown label in line: 1057 >3
               >3             CPX   SPROOT

Unknown label in line: 1057 >4
               >4             LDX   INDX
8CD7: 90 56    >5             BCC   :2
8CD9: A9 FF    >6             LDA   #$FF       Mark the current thread

Unknown label in line: 1057 >7
               >7             STA   TABOFT,X    before switching to another
8CDD: B0 65    >8             BCS   KX3        Always branch

Unknown label in line: 1057 >9
               >9    :2       BIT   INHACTV
8CE1: 30 7C    >10            BMI   RESTORD

Unknown label in line: 1057 >11
               >11            DEC   CTRACTV    Time for a context switch?
8CE5: D0 78    >12            BNE   RESTORD    Not yet

Unknown label in line: 1057 >13
               >13            LDA   TABOFT,X   Get BASIC array where to save
8CE9: 20 DE 8D >14            JSR   NEXTC2      content
8CEC: DA       >16            PHX
8CED: 20 EC 8D >18            JSR   SAVER      Perform the SAVE
8CF0: FA       >20            PLX              ;Get back the new context index
               >21   KX3
8CF1: 20 C5 8D >25            JSR   NEXTCTX    Search for a new context index
8CF4: 90 79    >26            BCC   RESTOR2    Found one
               >27   * Restore context from calling BASIC line
8CF6: 20 BC 8D >28   KILLEMAL JSR   SETLTR     Restore context from calling
8CF9: 20 A8 8D >29            JSR   RESTORC     BASIC line

Unknown label in line: 1057 >30
               >30            LDX   SPROOT
8CFE: 86 F8    >31            STX   REMSTK
8D00: 20 5B 8D >32            JSR   R0
8D03: 9A       >33            TXS
8D04: 4C D2 D7 >34            JMP   NEWSTT

Unknown label in line: 1057 >35
               >35   R0       LSR   MTACTV
8D09: 60       >36            RTS
               >37
8D0A: 20 E1 8F >38   RESTORD  JSR   LBS10
8D0D: 4C 20 D8 >39            JMP   $D820
```

```
                >40   * General purpose restore routine
                >41   * Input: X register index of context

Unknown label in line: 1057 >42
                >42   RESTOR1  LDA   TABOFT,X
8D12: C9 FF     >43            CMP   #$FF       Safe guard: do not restore a

Bad branch in line: 1057 >44
                >44            BEQ   RESTORF    terminated thread..
8D16: 20 DE 8D  >45            JSR   NEXTC2
                >46
                >47   * Input from caller: X: context index

Unknown label in line: 1057 >48
                >48   RESTOR2  LDA   ICTRACTV   Reinit counter

Unknown label in line: 1057 >49
                >49            STA   CTRACTV     value
                >50   * Update ITHREAD% variable value

Unknown label in line: 1057 >51
                >51            LDA   ITVADDR+1
8D1F: F0 65     >52            BEQ   RESTOR     Skip if no var. defined
8D21: 85 07     >53            STA   AUXPTR+1

Unknown label in line: 1057 >54
                >54            LDA   ITVADDR
8D25: 85 06     >55            STA   AUXPTR
8D27: 8A        >56            TXA
8D28: A0 03     >57            LDY   #3
8D2A: 91 06     >58            STA   (AUXPTR),Y
8D2C: 18        >59   RESTOR   CLC
8D2D: A0 1C     >60            LDY   #28        Trigger the page in routine if
8D2F: 20 07 8E  >61            JSR   SWPIO       defined

Unknown label in line: 1057 >63
                >63            LDX   INDX
8D34: B0 0E     >65            BCS   KX3
                >66   * Do the RESTOR itself
                >67   * Input: LOWTR: Array base address
8D36: 20 A8 8D  >68            JSR   RESTORC
                >69   * Do the Stack restore
8D39: A0 1F     >70            LDY   #31        From offset 31 within context
8D3B: A6 F8     >71            LDX   REMSTK      array storage
8D3D: 9A        >72   RESTORX  TXS

Unknown label in line: 1057 >73
                >73   ]LOOP    CPX   SPROOT     Until SPROOT value is reached
8D40: B0 1D     >74            BCS   RESTORD
8D42: E8        >75            INX
8D43: B1 9B     >76            LDA   (LOWTR),Y
8D45: 9D 00 01  >77            STA   $0100,X
8D48: C8        >78            INY
8D49: 90 F3     >79            BCC   ]LOOP      Always
8D4B: 60        >80   RESTORF  RTS
                >81
8D4C: 20 16 8E  >83   RESTORC  JSR   LBS06
```

```
8D4F: 90 02    >84            BCC   *+4
8D51: 85 D8    >85            STA   ERRFLG
8D53: B1 9B    >93   ]LOOP    LDA   (LOWTR),Y

Unknown label in line: 1057 >94
               >94            LDX   P0OFFSET-8,Y
8D57: 95 00    >95            STA   0,X
8D59: 88       >96            DEY
8D5A: E0 F8    >97            CPX   #REMSTK
8D5C: D0 F5    >98            BNE   ]LOOP
8D5E: 60       >99            RTS
               >100
               >101  * Subroutine to get the context storage index for
               >102  * global (i.e. Perrsoft MT kernel calling line)
8D5F: A9
Unknown label in line: 1057 >103
               >103  SETLTR   LDA   #SVPTR-8
8D62: 85 9B    >104           STA   LOWTR
8D64: A9
Unknown label in line: 1057 >105
               >105           LDA   #>SVPTR-8
8D67: 85 9C    >106           STA   LOWTR+1
8D69: 60       >107           RTS
               >108  * Subroutine to get the next context after the current one
               >109  * (index in X).
8D6A: A0 00    >110  NEXTCTX  LDY   #0         ctr. to avoid counting too far
8D6C: E8       >111  ]LOOP    INX             ;Wrap around the context ptr
8D6D: E0
Unknown label in line: 1057 >112
               >112           CPX   #TABOFT-TABOFB area..
8D70: 90 5C    >113           BCC   :0
8D72: A2 00    >114           LDX   #0         Perform wrap...

Unknown label in line: 1057 >115
               >115  :0       LDA   TABOFT,X
8D76: C9 FF    >116           CMP   #$FF       Got an active one (iif <> $FF)
8D78: D0 61    >117           BNE   :1         Yes...
8D7A: C8       >118           INY             ;Bump counter
8D7B: C0
Unknown label in line: 1057 >119
               >119           CPY   #TABOFT-TABOFB till all scanned
8D7E: 90 EC    >120           BCC   ]LOOP      Not yet: see next context ptr
8D80: 60       >121           RTS             ;Exit with carry set..

Unknown label in line: 1057 >122
               >122  :1       STX   INDX       Memorize the new context index
8D83: A8       >123  NEXTC2   TAY             ;From offset to absolute address

Unknown label in line: 1057 >124
               >124           LDA   TABOFB,X   by adding the ARYTAB base address
8D86: 65 6B    >125           ADC   ARYTAB     for arrays within Applesoft
8D88: 85 9B    >126           STA   LOWTR
8D8A: 98       >127           TYA
8D8B: 65 6C    >128           ADC   ARYTAB+1
8D8D: 85 9C    >129           STA   LOWTR+1    Result in LOWTR pointer..
8D8F: 60       >130           RTS             ;Exit with carry clear (always)
               >131
```

```
                >132  * Save the context into BASIC array
                >133  * Input: LOWTR: array base address
8D90: 20 23 8E  >134  SAVER    JSR   SAVERC
8D93: A0 1E     >135           LDY   #30        Possible trigger for page out
8D95: 20 07 8E  >136           JSR   SWPIO       event...
                >137  * Now it´s time to save the stack extension
8D98: A0 1F     >138           LDY   #31
                >139  * As a subroutine, do not depend on current stack ptr.
                >140  * But rather on memorized stack ptr. (within exec loop)
8D9A: A6 F8     >141           LDX   REMSTK

Unknown label in line: 1057 >142
                >142  ]LOOP    CPX   SPROOT
8D9E: B0 66     >143           BCS   :0
8DA0: E8        >144           INX
8DA1: BD 00 01  >145           LDA   $0100,X
8DA4: 91 9B     >146           STA   (LOWTR),Y
8DA6: C8        >147           INY
8DA7: 90 F3     >148           BCC   ]LOOP
8DA9: 60        >149  :0       RTS
                >150
                >151  * Routine to possibly trigger page in/page out routine
                >152  * for every configured coroutine. Inputs are:
                >153  * LOWTR: context array base address
                >154  * Y either 30 or 28 for page in/out event
8DAA: B1 9B     >155  SWPIO    LDA   (LOWTR),Y
8DAC: F0 67     >156           BEQ   :0         No routine defined
8DAE: 85 07     >157           STA   AUXPTR+1
8DB0: 88        >158           DEY
8DB1: B1 9B     >159           LDA   (LOWTR),Y
8DB3: 85 06     >160           STA   AUXPTR
                >161  * Called routine must preserve registers
8DB5: 6C 06 00  >162           JMP   (AUXPTR)
8DB8: 60        >163  :0       RTS
                >164
8DB9: A0 1A     >165  LBS06    LDY   #26
8DBB: B1 9B     >166  LBS061   LDA   (LOWTR),Y
8DBD: D0 61     >167           BNE   :0
8DBF: 38        >169           SEC
8DC0: A0
Unknown label in line: 1057 >171
                >171  :1       LDY   #PIOFFSET-P0OFFSET+8-1
8DC3: 60        >172           RTS
8DC4: 18        >174  :0       CLC
8DC5: 88        >178           DEY              ;Shortcut for
8DC6: 60        >179           RTS              ; LDY #PEOFFSET-P0OFFSET+8-1
                >180
8DC7: 20 16 8E  >182  SAVERC   JSR   LBS06

Unknown label in line: 1057 >187
                >187  ]LOOP    LDX   P0OFFSET-8,Y
8DCC: B5 00     >188           LDA   0,X        Value to save
8DCE: 91 9B     >189           STA   (LOWTR),Y
8DD0: 88        >190           DEY
8DD1: E0 F8     >191           CPX   #REMSTK
8DD3: D0 F5     >192           BNE   ]LOOP
8DD5: 60        >193           RTS
```

```
                1058
                1059            PUT   PEERMOUSTIME
                >1    * Base addresses for mouse interface
                >2    BAXLO     EQU   $0478       X low
                >3    BAYLO     EQU   $04F8       Y low
                >4    BAXHI     EQU   $0578       X high
                >5    BAYHI     EQU   $05F8       Y high
                >6    BAMBS     EQU   $0778       Button status
                >7
                >8    TRACE     EQU   $D805
                >9    IRQV      EQU   $03FE       Page 3 Interrupt vector
                >10
                >11   * Reason codes for entering Mouse interface
                >12   RSETM     =     0
                >13   RSRVM     =     1
                >14   RREAD     =     2
                >15   RCLR      =     3
                >16   RPOS      =     4
                >17   RCLM      =     5
                >18   RHOM      =     6
                >19   RINI      =     7
                >20
                >21   CONINT    EQU   $E6FB       FAC to single byte
                >22
                >23   * Interrupt servicing routine
8DD6: A2 01     >24   IRQHDLR   LDX   #RSRVM
8DD8: 20 BD 90  >25             JSR   TOMOUSE

Bad branch in line: 1059 >26
                >26             BCS   :2          ; Not from mouse or spurious

Unknown label in line: 1059 >27
                >27             LDX   MOSL
8DDF: BD 78 07  >28             LDA   BAMBS,X
8DE2: 4A        >29             LSR
                >30   * Movement interrupt bit into b0 and
                >31   * button bit into b1, VBL interrupt bit
                >32   * into b2
8DE3: 29 07     >33             AND   #7          mask out other bits
8DE5: AA        >34             TAX

Unknown label in line: 1059 >35
                >35             LDA   MSTATUS,X   Get internal status

Unknown label in line: 1059 >36
                >36             STA   WORKPL1
8DEA: A2 02     >37             LDX   #RREAD
8DEC: 20 BD 90  >38             JSR   TOMOUSE

Unknown label in line: 1059 >39
                >39             BIT   WORKPL1
8DF1: 10 7A     >40             BPL   :1
                >41   * Decrement runtime counter
8DF3: 18        >43             CLC
8DF4: FB        >44             XCE
8DF5: C2 20     >45             REP   $20         16bits for mem/accu
```

```
Unknown label in line: 1059 >46
                >46             DEC   TIINC
8DF9: D0 6A     >47             BNE   :03
8DFB: 08        >48             PHP                 ;Z bit on stack

Unknown label in line: 1059 >49
                >49             LDA   KTINC

Unknown label in line: 1059 >50
                >50             STA   TIINC
8E00: 28        >51             PLP                 ;restore Z bit from stack
8E01: FB        >52   :03       XCE                 ;Back to emulation mode
8E02: F0 69     >53             BEQ   :1            ;RTI if time for advising
8E04: A9 80     >66             LDA   #$80

Unknown label in line: 1059 >67
                >67             TRB   WORKPL1

Unknown label in line: 1059 >73
                >73   :1        LDA   WORKPL1

Unknown label in line: 1059 >75
                >75             TSB   MIRQST
8E0C: 40        >80   ]LOOP     RTI
                >81
                >82   * No spurious interrupt is fatal to us..
                >83   * I´m afraid of no ghosts.... ;-)

Unknown label in line: 1059 >84
                >84   :2        LDA   OLDVECT+1
8E0F: C9 FF     >85             CMP   #>$FF65
8E11: D0 6F     >86             BNE   :20

Unknown label in line: 1059 >87
                >87             LDA   OLDVECT
8E15: C9 65     >88             CMP   #$FF65
8E17: F0 F3     >89             BEQ   ]LOOP

Unknown label in line: 1059 >90
                >90   :20       JMP   (OLDVECT)
                >91
                >104
                >105  * Install new IRQ handler and save the original handler
                >106  * to build a daisy chain..
                >107  * Nouveau mode dans MOMODE

Unknown label in line: 1059 >108
                >108  INSIRQV   LDA   MOMODE
8E1D: C9 02     >109            CMP   #2

Bad branch in line: 1059 >110
                >110            BCC   :1
8E21: 78        >112            SEI
8E22: 18        >113            CLC
8E23: FB        >114            XCE
8E24: C2 20     >115            REP   $20
                >116            MX    %01
```

```
8E26: AD FE 03 >117           LDA   IRQV       Si IRQV est deja egal a IRQHDLR
8E29: C9 33 8E >118           CMP   #IRQHDLR   alors pas de m.a.j. necessaire
8E2C: F0 74    >119           BEQ   :0

Unknown label in line: 1059 >120
               >120           STA   OLDVECT
8E30: A9 33 8E >121           LDA   #IRQHDLR
8E33: 8D FE 03 >122           STA   IRQV
8E36: 38       >123  :0       SEC
8E37: FB       >124           XCE
               >125           MX    %11
8E38: 58       >138           CLI
8E39: 60       >139  :1       RTS
               >140
               >141  * Deinstall IRQ handler

Unknown label in line: 1059 >142
               >142  DINSIRQV LDA   MOMODE
8E3C: C9 02    >143           CMP   #2
8E3E: B0 7F    >144           BCS   :1
8E40: 78       >145           SEI
8E41: 18       >147           CLC              ;This instruction...
8E42: FB       >148           XCE
8E43: C2 20    >149           REP   $20
               >150           MX    %01

Unknown label in line: 1059 >151
               >151           LDA   OLDVECT
8E47: F0 74    >152           BEQ   :0

Unknown label in line: 1059 >153
               >153           STZ   OLDVECT
8E4B: 8D FE 03 >154           STA   IRQV
8E4E: 38       >155  :0       SEC
8E4F: FB       >156           XCE
               >157           MX    %11
8E50: 60       >171  :1       RTS
               >172
8E51: 48       >173  CMPCLAMP PHA
               >174  * X/Y min% expression
8E52: 20 8C 8F >175           JSR   NEVAL
8E55: 8D 78 05 >176           STA   $0578
8E58: 8C 78 04 >177           STY   $0478
               >178  * X/Y max% expression
8E5B: 20 8C 8F >179           JSR   NEVAL
8E5E: 8D F8 05 >180           STA   $05F8
8E61: 8C F8 04 >181           STY   $04F8
8E64: 68       >182           PLA
8E65: A2 05    >183           LDX   #RCLM
8E67: 4C BD 90 >184           JMP   TOMOUSE
               >185
8E6A: C5 A1    >186  IVALARG  CMP   FAC+4
8E6C: 90 01    >187           BCC   *+3
8E6E: 60       >188           RTS
8E6F: 68       >189           PLA
8E70: 68       >190           PLA
8E71: 4C 99 E1 >191  ]ERR     JMP   $E199      Illegal quantity error
```

```
                >192
8E74: A9 00     >193  COMCLAMP LDA   #0
8E76: 20 C0 8E  >194           JSR   CMPCLAMP
8E79: A9 01     >195           LDA   #1
8E7B: D0 43     >196           BNE   CMPCLAMP
                >197
8E7D: 20 F2 8A  >198  ROUT10   JSR   NCHKCOM
8E80: 20 1C 8B  >199           JSR   NGETBYT    Get reason code in X reg.
8E83: CA        >200           DEX
8E84: CA        >201           DEX
8E85: 30 EA     >202           BMI   ]ERR
8E87: E0 05     >203           CPX   #5
8E89: B0 E6     >204           BCS   ]ERR
8E8B: 20 41 92  >205           JSR   ISMOUSH

Unknown label in line: 1059 >206
                >206           LDA   MOMODE
8E90: 29 0F     >207           AND   #$F
8E92: D0 75     >208           BNE   :1
8E94: A2 25     >209           LDX   #37
8E96: 4C 50 92  >210           JMP   NERRH
                >211  * Only READ (2), CLEAR (3), POS(4), CLAMP (5) and HOME (6)
                >212  * reason codes are valid.
8E99: 8A        >213  :1       TXA

Bad branch in line: 1059 >214
                >214           BEQ   COMREAD
8E9C: CA        >215           DEX
8E9D: F0 79     >216           BEQ   COMCLEAR
8E9F: CA        >217           DEX

Bad branch in line: 1059 >218
                >218           BEQ   COMPOS
8EA2: CA        >219           DEX
8EA3: F0 3E     >220           BEQ   COMCLAMP
8EA5: A2 06     >221           LDX   #RHOM
8EA7: 2C        >222           HEX   2C         Skip next two bytes
8EA8: A2 CE     >223  COMCLEAR LDX   #RCLEAR
8EAA: 4C BD 90  >224  FINMOUSE JMP   TOMOUSE
                >225

Unknown label in line: 1059 >226
                >226  COMREAD  LDX   MODERUN
8EAF: D0 76     >227           BNE   :1
8EB1: A2 02     >228           LDX   #RREAD
8EB3: 20 BD 90  >229           JSR   TOMOUSE
                >230  * Handles X% host variable

Unknown label in line: 1059 >231
                >231  :1       LDX   MOSL
8EB8: BD 78 05  >232           LDA   BAXHI,X
8EBB: 20 67 8F  >233           JSR   NPTRG
8EBE: BD 78 04  >234           LDA   BAXLO,X
8EC1: 91 83     >235           STA   (VARPNT),Y
                >236  * Handle Y% host variable
8EC3: BD F8 05  >237           LDA   BAYHI,X
8EC6: 20 67 8F  >238           JSR   NPTRG
```

```
8EC9: BD F8 04 >239           LDA   BAYLO,X
8ECC: 91 83    >240           STA   (VARPNT),Y
               >241  * Handle S% for button status variable
8ECE: A9 00    >242           LDA   #0
8ED0: 20 67 8F >243           JSR   NPTRG
8ED3: BD 78 07 >244           LDA   BAMBS,X
8ED6: 91 83    >245           STA   (VARPNT),Y
8ED8: 60       >246           RTS
               >247
               >248  COMPOS
               >249  * X% expression
8ED9: 20 8C 8F >250           JSR   NEVAL
8EDC: 9D 78 05 >251           STA   BAXHI,X
8EDF: 98       >252           TYA
8EE0: 9D 78 04 >253           STA   BAXLO,X
               >254  * Y% expression
8EE3: 20 8C 8F >255           JSR   NEVAL
8EE6: 9D F8 05 >256           STA   BAYHI,X
8EE9: 98       >257           TYA
8EEA: 9D F8 04 >258           STA   BAYLO,X
8EED: A2 04    >259           LDX   #RPOS
8EEF: 4C 1A 8F >260           JMP   FINMOUSE
               >261
8EF2: 4C 76 DD >262  ]ERR     JMP   GOTMIERR   TYPE MISMATCH ERROR
8EF5: 48       >263  NPTRG    PHA
8EF6: 20 F2 8A >264           JSR   NCHKCOM
8EF9: 20 8E 7E >265           JSR   NPTRGTX
8EFC: A5 12    >266           LDA   INTTYP
8EFE: 10 F2    >267           BPL   ]ERR
8F00: 29 0F    >268           AND   #15        cater for integer subtypes
8F02: F0 76    >269           BEQ   :1         only $80 and $82 are valid
8F04: C9 02    >270           CMP   #2
8F06: D0 EA    >271           BNE   ]ERR

Unknown label in line: 1059 >272
               >272  :1       LDX   MOSL
8F0A: 68       >273           PLA
8F0B: 92 83    >275           STA   (VARPNT)
8F0D: A0 01    >276           LDY   #1
8F0F: 60       >282           RTS
               >283
               >284  * Result in FAC+3, FAC+4
8F10: 20 F2 8A >285  NEVALC   JSR   NCHKCOM
8F13: 20 8C 89 >286           JSR   NFRMNUM
8F16: 4C D6 7C >287           JMP   NROUT      Replac. for ROUND.FAC/AYINT
               >288
8F19: 20 83 8F >289  NEVAL    JSR   NEVALC
8F1C: A5 A0    >290           LDA   FAC+3
8F1E: A4 A1    >291           LDY   FAC+4

Unknown label in line: 1059 >292
               >292           LDX   MOSL
8F22: 60       >293  ]RET     RTS
               >294
               >295  * Common subroutine for parsing new tokens
               >296  * X upon entry: 0: updates TXTPTR if token found
               >297  * 1: skip updating TXTPTR even when token found
```

```
8F23: 86 C0    >298  COMLBS   STX   GFLAG
8F25: B2 B8    >300           LDA   (TXTPTR)

Bad branch in line: 1059 >305
               >305           BMI   :2
8F29: C9 4D    >306           CMP   #´M´
8F2B: F0 78    >307           BEQ   :1
8F2D: C9 54    >308           CMP   #´T´

Bad branch in line: 1059 >309
               >309           BNE   :2
8F31: A2 03    >310  :1       LDX   #3
8F33: 20 15 87 >311           JSR   RECON1
8F36: F0 EA    >312           BEQ   ]RET
8F38: 20 A3 90 >313           JSR   COMINT4    Check mouse hardware/reinit
8F3B: A6 C0    >314           LDX   GFLAG
8F3D: D0 E3    >315           BNE   ]RET
8F3F: 4C 98 D9 >316           JMP   ADDON      will exit with Z flag clear
               >317  :2
8F42: A2 00    >319           LDX   #0
8F44: 60       >323  ]RET     RTS
               >324
               >325  * New instructions handling
               >326  * for MOUSE and TIMER instructions
8F45: 4C CC 7A >327  ]LOOP    JMP   RST102
8F48: 68       >328  ]ERR1    PLA             ;Pull IDMOCL from stack
8F49: 68       >329           PLA             ;Pull return address
8F4A: 68       >330           PLA
8F4B: 4C C9 DE >331  ]ERR     JMP   SYNERR
               >332  * MOUSE/TIMER STOP handler
8F4E: C0
Unknown label in line: 1059 >333
               >333  ]JLOOP   CPY   #OFFTIM-TOFFST
8F51: A2 00    >334           LDX   #0
8F53: 90 01    >335           BCC   *+3        Branch iif MOUSE
8F55: E8       >336           INX

Unknown label in line: 1059 >337
               >337           LDA   MOMODE

Unknown label in line: 1059 >338
               >338           AND   MOETMSK,X
               >339  * Compare to minimum allowable value

Unknown label in line: 1059 >340
               >340           CMP   MOCMPVAL,X
8F5C: B0 7B    >341           BCS   :0         OK iif greater or equal
8F5E: A2 25    >342           LDX   #37
8F60: 4C 50 92 >343           JMP   NERRH
8F63: A9 01    >344  :0       LDA   #1         Update MODEPEC configuration

Unknown label in line: 1059 >345
               >345           STA   MODEPEC,X
8F67: 4C D2 D7 >346           JMP   NEWSTT
8F6A: A2 00    >347  LBS10    LDX   #0
8F6C: 20 97 8F >348           JSR   COMLBS
8F6F: F0 D4    >349           BEQ   ]LOOP
```

```
8F71: A5 BD    >350           LDA   IDMOCL
8F73: 48       >351           PHA
8F74: B2 B8    >353           LDA   (TXTPTR)
8F76: A0 01    >354           LDY   #1
8F78: C9 B3    >360           CMP   #$B3       STOP token?

Bad branch in line: 1059 >361
               >361           BEQ   :3
8F7C: C9 B4    >362           CMP   #$B4

Bad branch in line: 1059 >363
               >363           BEQ   :3         ON token?
8F80: C9 4F    >364           CMP   #´O´
8F82: D0 C4    >365           BNE   ]ERR1
8F84: A2 05    >366           LDX   #5         Look up possible OFF pattern
8F86: 20 15 87 >367           JSR   RECON1
8F89: F0 BD    >368           BEQ   ]ERR1
8F8B: AA       >369  :3       TAX              ;X STOP/ON token or 0 (OFF)
8F8C: 86 B4    >370           STX   XSAV
8F8E: 20 98 D9 >371           JSR   ADDON
8F91: 7A       >372           PLY
8F92: 68       >373           PLA
8F93: 68       >374           PLA
8F94: 20 CC 7A >375           JSR   RST102

Bad branch in line: 1059 >376
               >376           BEQ   :23        If EOI found
8F99: E0 B4    >377           CPX   #$B4
8F9B: D0 AE    >378           BNE   ]ERR       SYNTAX ERR if not ON nor EOI
8F9D: DA       >379           PHX
8F9E: 5A       >380           PHY
8F9F: 20 83 8F >381           JSR   NEVALC     Get factor/mode value after comma
8FA2: 7A       >382           PLY
8FA3: FA       >383           PLX
8FA4: 86 B4    >384           STX   XSAV
8FA6: C0
Unknown label in line: 1059 >385
               >385           CPY   #OFFMOU-TOFFST
8FA9: D0 7C    >386           BNE   :20
8FAB: 20 FE E6 >387           JSR   $E6FE      FAC integer -> single byte
8FAE: 2C       >388           HEX   2C
8FAF: A2 01    >389  :23       LDX   #1
8FB1: 86 C0    >390  :20       STX   GFLAG
8FB3: 84 BD    >391           STY   IDMOCL
8FB5: A5 B4    >392           LDA   XSAV        A: ON/OFF/STOP index
8FB7: C9 B3    >393           CMP   #$B3       STOP token?
8FB9: F0 93    >394           BEQ   ]JLOOP
               >395  * IDMOCL in page zero, STOP/ON/OFF indic. in A reg.
8FBB: A6 BD    >396           LDX   IDMOCL
8FBD: E0
Unknown label in line: 1059 >397
               >397           CPX   #OFFMOU-TOFFST

Bad branch in line: 1059 >398
               >398           BNE   TIMEINST
               >399
               >400  * Mouse event handler
```

```
8FC2: C9 B4    >401           CMP   #$B4       MOUSE ON?
8FC4: D0 04    >402           BNE   *+6        No
8FC6: A2 00    >403           LDX   #0

Bad branch in line: 1059 >404
               >404           BEQ   :8
8FCA: A2 07    >405           LDX   #7
8FCC: E4 C0    >406  ]LOOP    CPX   GFLAG
8FCE: F0 7C    >407           BEQ   :8
8FD0: CA       >408           DEX
8FD1: CA       >409           DEX
8FD2: 10 F8    >410           BPL   ]LOOP
8FD4: 4C 4E 92 >411  ]LOOP    JMP   NILLM
8FD7: A9 07    >413  :8       LDA   #7

Unknown label in line: 1059 >414
               >414           TRB   MOMODE
8FDB: 8A       >415           TXA

Unknown label in line: 1059 >416
               >416           TSB   MOMODE
8FDE: C9 02    >417           CMP   #2
8FE0: A9 00    >426           LDA   #0
8FE2: A8       >427           TAY
8FE3: 90 02    >428           BCC   *+4
8FE5: A9 02    >429  COMMON9  LDA   #2

Unknown label in line: 1059 >430
               >430           STA   MODEPEC,Y

Unknown label in line: 1059 >431
               >431  COMMON   LDA   MOMODE
8FEB: 48       >432           PHA
8FEC: 20 85 8E >433           JSR   INSIRQV
8FEF: 68       >434           PLA
8FF0: A2 00    >435           LDX   #RSETM
8FF2: 20 BD 90 >436           JSR   TOMOUSE
8FF5: B0 DD    >437           BCS   ]LOOP
8FF7: 20 A6 8E >438           JSR   DINSIRQV
8FFA: 4C D2 D7 >439           JMP   NEWSTT
               >440
8FFD: C9 B4    >441  TIMEINST CMP   #$B4       TIMER ON
8FFF: A9 08    >443           LDA   #8

Unknown label in line: 1059 >444
               >444           TRB   MOMODE
9003: 90 5C    >445           BCC   COMMON

Unknown label in line: 1059 >446
               >446           TSB   MOMODE
9007: 24 C0    >456           BIT   GFLAG
9009: 30 06    >457           BMI   *+8
900B: A2 01    >458           LDX   #1
900D: A0 00    >459           LDY   #0
900F: 10 04    >460           BPL   *+6        Always
9011: A6 A1    >461           LDX   FAC+4
9013: A4 A0    >462           LDY   FAC+3
```

```
9015: 08       >463           PHP
9016: 78       >464           SEI

Unknown label in line: 1059 >465
               >465           STY   KTINC+1

Unknown label in line: 1059 >466
               >466           STX   KTINC

Unknown label in line: 1059 >467
               >467           STY   TIINC+1

Unknown label in line: 1059 >468
               >468           STX   TIINC
901F: 28       >469           PLP
9020: A0 01    >470           LDY   #1
9022: B0 38    >471           BCS   COMMON9    Always
               >472
               >473  * Do we have suitable mouse hardware?
9024: 20 41 92 >474  COMINT4  JSR   ISMOUSH    Fall into SWREINIT if yes
               >475  * Routine below to check whether we should init the
               >476  * MOUSE system?
               >477  SWREINIT
9027: A9 80    >479           LDA   #$80

Unknown label in line: 1059 >480
               >480           TSB   MONU

Bad branch in line: 1059 >481
               >481           BNE   :0
               >488  * INITMOUSE was performed on Peersoft boot when in an
               >489  * Apple 2,2+ host.

Unknown label in line: 1059 >490
               >490           LDA   MACHINE

Bad branch in line: 1059 >491
               >491           BEQ   :0
9031: 5A       >492           PHY
9032: A2 07    >493           LDX   #RINI
9034: 20 BD 90 >494           JSR   TOMOUSE
9037: 7A       >495           PLY
9038: 60       >496  :0       RTS
               >497

Unknown label in line: 1059 >498
               >498  ]LOOP    JMP   (MVECTOR)
               >499

Unknown label in line: 1059 >500
               >500  TOMOUSE  LDY   OM_DEB,X

Unknown label in line: 1059 >501
               >501           LDX   MOCN
903F: 08       >502           PHP
9040: 78       >503           SEI
```

```
Unknown label in line: 1059 >504
               >504           STY   MVECTOR

Unknown label in line: 1059 >505
               >505           LDY   MON0
9045: 20 39 90 >506           JSR   ]LOOP
9048: B0 03    >507           BCS   *+5
904A: 28       >508           PLP
904B: 18       >509           CLC
904C: 60       >510           RTS
904D: 28       >511           PLP
904E: 38       >512           SEC
904F: 60       >513           RTS
               >514
               >515  * Entry routine for MOUSE functions (either MOUSE or
               >516  * TIMER).
9050: 48       >517  MTFUNC   PHA
9051: 20 FB E6 >518           JSR   CONINT
9054: 20 EF 8A >519           JSR   NCHKCLS
9057: 20 A3 90 >520           JSR   COMINT4
905A: 68       >521           PLA

Bad branch in line: 1059 >522
               >522           BNE   TFUNC
905D: A9 02    >523           LDA   #2
905F: 20 D9 8E >524           JSR   IVALARG

Unknown label in line: 1059 >525
               >525           LDX   MODERUN
9064: D0 05    >526           BNE   *+7        Branch if within interrupt
9066: A2 02    >527           LDX   #RREAD
9068: 20 BD 90 >528           JSR   TOMOUSE

Unknown label in line: 1059 >529
               >529           LDX   MOSL
906D: A5 A1    >531           LDA   FAC+4
906F: 3A       >532           DEC

Bad branch in line: 1059 >537
               >537           BPL   :1
9072: BD 78 05 >538           LDA   BAXHI,X    MOUSE(0) means read X
9075: BC 78 04 >539           LDY   BAXLO,X
9078: 4C F2 E2 >540  ]LOOP    JMP   GIVAYF
               >541  :1       DO    KOPT-K6502
907B: 3A       >542           DEC

Bad branch in line: 1059 >546
               >546           BPL   :2
907E: BD F8 05 >547           LDA   BAYHI,X    MOUSE(1) means read Y
9081: BC F8 04 >548           LDY   BAYLO,X
9084: 80 F2    >550           BRA   ]LOOP
9086: BC 78 07 >554  :2       LDY   BAMBS,X    MOUSE(2) means read buttons
9089: 4C 01 E3 >555           JMP   SNGFLT
908C: A9 01    >556  TFUNC    LDA   #1
908E: 20 D9 8E >557           JSR   IVALARG
9091: 20 39 92 >558           JSR   ISHOSTOK
9094: A2 00    >559           LDX   #0
```

```
9096: A5 A1    >560           LDA   FAC+4
9098: F0 02    >561           BEQ   *+4
909A: A2 02    >562           LDX   #2

Unknown label in line: 1059 >563
               >563           LDA   KTINC+1,X

Unknown label in line: 1059 >564
               >564           LDY   KTINC,X
90A0: 80 D6    >566           BRA   ]LOOP
               >570
               >571  * Desactive le traitement d´une interruption (sur RETURN)
               >572  * Y en entree: indice de l´interruption
90A2: A9 00    >573  COMINT1  LDA   #0

Unknown label in line: 1059 >574
               >574           STA   MODERUN,Y
90A6: 3A       >576           DEC

Unknown label in line: 1059 >580
               >580           STA   YICUR
               >581  * MODEPEC passe de STOP a ON

Unknown label in line: 1059 >583
               >583           LDA   MODEPEC,Y
90AB: C9 01    >584           CMP   #1

Bad branch in line: 1059 >585
               >585           BNE   :0
90AF: 1A       >586           INC

Unknown label in line: 1059 >594
               >594           STA   MODEPEC,Y

Unknown label in line: 1059 >595
               >595  :0       LDA   TPT_B,Y
90B4: 85 B8    >596           STA   TXTPTR

Unknown label in line: 1059 >597
               >597           LDA   TPT_T,Y
90B8: 85 B9    >598           STA   TXTPTR+1

Unknown label in line: 1059 >599
               >599           LDA   CLN_B,Y
90BC: 85 75    >600           STA   CURLIN

Unknown label in line: 1059 >601
               >601           LDA   CLN_T,Y
90C0: 85 76    >602           STA   CURLIN+1

Unknown label in line: 1059 >603
               >603           LDA   OTPT_B,Y
90C4: 85 79    >604           STA   OLDTEXT

Unknown label in line: 1059 >605
               >605           LDA   OTPT_T,Y
90C8: 85 7A    >606           STA   OLDTEXT+1
```

```
Unknown label in line: 1059 >607
               >607           LDX   SVMTACTV

Unknown label in line: 1059 >608
               >608           LDA   MODERUN

Unknown label in line: 1059 >609
               >609           ORA   MODERUN+1
90D0: D0 06    >610           BNE   *+8

Unknown label in line: 1059 >611
               >611           STA   SVMTACTV

Unknown label in line: 1059 >612
               >612           STX   MTACTV
90D6: A0 05    >613           LDY   #5

Unknown label in line: 1059 >614
               >614           CPY   FRGNDCTX

Bad branch in line: 1059 >615
               >615           BNE   :1
90DC: 68       >616           PLA
90DD: 68       >617           PLA
90DE: 4C A8 92 >618           JMP   RW2
90E1: 60       >619  :1       RTS
               >620
               >621  * Routine en charge de determiner si l´interruption peut
               >622  * ou non etre cascadee.
               >623  * Sortie: bitN a 0 ssi possibilite de cascade (indice
               >624  * dans Y)
90E2: A0 01    >625  COMINT2  LDY   #1         On commence par la TIMER

Unknown label in line: 1059 >626
               >626  ]LOOP    LDA   MSKINT,Y
90E6: 08       >627           PHP              ;Sauve le interrupt enable
90E7: 78       >628           SEI              ;courant

Unknown label in line: 1059 >629
               >629           AND   MIRQST

Bad branch in line: 1059 >630
               >630           BEQ   :3
               >631  * Uniquement si prise en compte immediate..

Unknown label in line: 1059 >632
               >632           LDX   MODEPEC,Y
90EE: E0 02    >633           CPX   #2

Bad branch in line: 1059 >634
               >634           BNE   :3
               >635  * Uniquement si routine non deja active

Unknown label in line: 1059 >636
               >636           LDX   MODERUN,Y
```

```
Bad branch in line: 1059 >637
               >637           BNE   :3

Unknown label in line: 1059 >639
               >639           TRB   MIRQST
90F8: 28       >646           PLP
90F9: A9 02    >647           LDA   #3-1       because from within a called subr
.
90FB: 20 D6 D3 >648           JSR   CHKMEM

Unknown label in line: 1059 >649
               >649           STY   YICUR

Unknown label in line: 1059 >650
               >650           LDA   MTACTV

Unknown label in line: 1059 >651
               >651           STA   SVMTACTV
9104: A9 01    >652           LDA   #1

Unknown label in line: 1059 >653
               >653           STA   MODEPEC,Y

Unknown label in line: 1059 >654
               >654           STA   MODERUN,Y
910A: 60       >655           RTS
910B: 28       >656  :3       PLP
910C: 88       >657           DEY
910D: 10 D5    >658           BPL   ]LOOP
910F: 60       >659           RTS
               >660
               >661  * Retour d´une interruption souris
9110: A0 00    >662  RETOURM  LDY   #0
9112: 2C       >663           HEX   2C         Skip next two bytes
9113: A0 01    >664  RETOURT  LDY   #1
9115: BA       >665           TSX
9116: 86 F8    >666           STX   REMSTK
9118: 20 2C 91 >667           JSR   COMINT1
911B: 20 5E 86 >668           JSR   DECTPTR
911E: 20 58 D8 >669           JSR   ISCNTC
9121: 4C 05 D8 >670           JMP   TRACE
               >671

Unknown label in line: 1059 >672
               >672  RNEWINST LDA   MODERUN

Unknown label in line: 1059 >673
               >673           ORA   MODERUN+1

Bad branch in line: 1059 >674
               >674           BEQ   RNI2
               >675  * Y a la bonne valeur selon MOUSE ou TIMER actifs

Unknown label in line: 1059 >676
               >676           LDY   YICUR

Bad branch in line: 1059 >677
```

```
                >677            BPL   :1
912E: C8        >678            INY             ;Y passe de FF a 0

Unknown label in line: 1059 >679
                >679            LDA   MODERUN+1
9131: F0 01     >680            BEQ   *+3
9133: C8        >681            INY             ;Y passe a 1

Unknown label in line: 1059 >682
                >682            STY   YICUR
9136: BA        >683  :1        TSX
9137: 8A        >684            TXA
                >685  * Routine terminee par RETURN/POP ayant ramene le SP

Unknown label in line: 1059 >686
                >686            CMP   INTSPTR,Y

Bad branch in line: 1059 >687
                >687            BCC   RNI2
913C: 20 2C 91  >688            JSR   COMINT1
                >689  * ...

Unknown label in line: 1059 >690
                >690  RNI2      LDA   MIRQST

Bad branch in line: 1059 >691
                >691            BEQ   :4
9143: 20 7C 91  >692            JSR   COMINT2

Bad branch in line: 1059 >693
                >693            BMI   :4        ;
                >694  * Reminder of current stack pointer
9148: BA        >695            TSX
9149: 8A        >696            TXA

Unknown label in line: 1059 >697
                >697            STA   INTSPTR,Y
                >698  * Builds the GOSUB stack frame
914C: C0 01     >699            CPY   #1        carry set iif TIMER int.
914E: B0 05     >701            BCS   *+7
9150: F4 B3 91  >702            PEA   RETOURM-1
9153: 90 03     >703            BCC   *+5
9155: F4 B6 91  >704            PEA   RETOURT-1
9158: A5 B9     >715            LDA   TXTPTR+1

Unknown label in line: 1059 >716
                >716            STA   TPT_T,Y
915C: 48        >717            PHA
915D: A5 B8     >718            LDA   TXTPTR

Unknown label in line: 1059 >719
                >719            STA   TPT_B,Y
9161: 48        >720            PHA
9162: A5 76     >721            LDA   CURLIN+1

Unknown label in line: 1059 >722
                >722            STA   CLN_T,Y
```

```
9166: 48       >723           PHA
9167: A5 75    >724           LDA   CURLIN

Unknown label in line: 1059 >725
               >725           STA   CLN_B,Y
916B: 48       >726           PHA
916C: A5 79    >727           LDA   OLDTEXT

Unknown label in line: 1059 >728
               >728           STA   OTPT_B,Y
9170: A5 7A    >729           LDA   OLDTEXT+1

Unknown label in line: 1059 >730
               >730           STA   OTPT_T,Y
9174: A9 B0    >731           LDA   #TOKGOSUB
9176: 48       >732           PHA
               >733  * and initialize the context for irq handler
               >734  * (before falling into NEWSTT)

Unknown label in line: 1059 >735
               >735           LDX   AHNDHI,Y

Unknown label in line: 1059 >736
               >736           LDA   AHNDLO,Y
917B: 85 B8    >737           STA   TXTPTR
917D: 86 B9    >738           STX   TXTPTR+1
917F: 4C D2 D7 >739           JMP   NEWSTT
               >740
9182: 4C 1A 8D >741  :4       JMP   RNEWISUI
               >742

Unknown label in line: 1059 >743
               >743  ISHOSTOK LDA   MACHINE
9187: C9 41    >744           CMP   #$41       Enhanced 2e ROM pattern

Bad branch in line: 1059 >745
               >745           BCC   HNOK
918B: 60       >746  ]RET     RTS

Unknown label in line: 1059 >747
               >747  ISMOUSH  LDA   MOCN
918E: D0 FB    >748           BNE   ]RET
9190: A2 20    >749           LDX   #32
9192: 2C       >750           HEX   2C         Skip next two byte
9193: A2 21    >751  HNOK     LDX   #33
9195: 68       >752  NERRHP   PLA              ;Pull return address
9196: 68       >753           PLA
9197: 2C       >754           HEX   2C
9198: A2 24    >755  NILLM    LDX   #36
               >756  * Error handler for new reason codes
               >757  * Upon entry, possible values of X
               >758  * 32: MOUSE NOT DETECTED
               >759  * UNSUPPORTED HARDWARE CONFIG.
               >760  * UNKNOWN APPLESOFT MOUSE EVENT HANDLER
               >761  * Same for TIMER
               >762  * ILLEGAL MOUSE MODE
               >763  * ILLEGAL MOUSE OP.
```

```
919A: 24 D8    >764  NERRH    BIT   ERRFLG
919C: 10 03    >765           BPL   *+5
919E: 4C F9 E2 >766           JMP   $E2F9      to ROM Error handler code
91A1: 20 FB DA >767           JSR   CRDO
91A4: 20 5A DB >768           JSR   $DB5A      Output question mark

Unknown label in line: 1059 >769
               >769           LDA   CODR-32,X
91A9: AA       >770           TAX

Unknown label in line: 1059 >771
               >771  ]LOOP    LDA   MESSERR,X
91AC: 48       >772           PHA
91AD: 20 5C DB >773           JSR   OUTDO
91B0: E8       >774           INX
91B1: 68       >775           PLA
91B2: 10 F6    >776           BPL   ]LOOP
91B4: 4C 2A D4 >777           JMP   $D42A      Fall into ROM code tail
               >778
91B7: 20 46 E7 >779  RWAIT    JSR   $E746      Get address in LINNUM,
91BA: 86 85    >780           STX   FORPNT      mask in X (saved)
91BC: A2 00    >781           LDX   #0
91BE: 20 B7 00 >782           JSR   $00B7
91C1: F0 03    >783           BEQ   *+5
91C3: 20 4C E7 >784           JSR   COMBYTE
91C6: 86 86    >785           STX   FORPNT+1
               >789  COMWAIT

Unknown label in line: 1059 >790
               >790  ]LOOP    LDA   MIRQST

Bad branch in line: 1059 >791
               >791           BNE   :2
91CC: B2 50    >793           LDA   (LINNUM)
91CE: 45 86    >797           EOR   FORPNT+1
91D0: 25 85    >798           AND   FORPNT
91D2: F0 F4    >799           BEQ   ]LOOP
91D4: 60       >800           RTS
91D5: 20 7C 91 >801  :2       JSR   COMINT2
91D8: 30 EE    >803           BMI   ]LOOP
91DA: 5A       >809           PHY
91DB: A0 05    >810           LDY   #5

Unknown label in line: 1059 >811
               >811           STY   FRGNDCTX

Unknown label in line: 1059 >812
               >812  ]LOOP    LDX   SVWOF,Y
91E1: B5 00    >813           LDA   0,X

Unknown label in line: 1059 >814
               >814           STA   SVA,Y
91E5: 88       >815           DEY
91E6: 10 F7    >816           BPL   ]LOOP
91E8: 7A       >817           PLY
91E9: 4C F3 91 >818           JMP   RNI2+10
               >819
```

```
91EC: A0 06    >820  RW2      LDY   #6

Unknown label in line: 1059 >821
               >821  ]LOOP    LDX   SVWOF-1,Y

Unknown label in line: 1059 >822
               >822           LDA   SVA-1,Y
91F2: 95 00    >823           STA   0,X
91F4: 88       >824           DEY
91F5: D0 F7    >825           BNE   ]LOOP

Unknown label in line: 1059 >826
               >826           STY   FRGNDCTX

Bad branch in line: 1059 >827
               >827           BEQ   COMWAIT    Always
                1060
91FB: A9
Unknown label in line: 1061
                1061 GN32768  LDA   #NEG32768
91FE: A0
Unknown label in line: 1062
                1062          LDY   #>NEG32768
9201: 60        1063          RTS
9202: A9
Unknown label in line: 1064
                1064 GP32768  LDA   #POS32768
9205: A0
Unknown label in line: 1065
                1065          LDY   #>POS32768
9208: 60        1066          RTS
                1067
9209: A9
Unknown label in line: 1068
                1068 GN65536  LDA   #NEG65536
920C: A0
Unknown label in line: 1069
                1069          LDY   #>NEG65536
920F: 60        1070          RTS
9210: A9
Unknown label in line: 1071
                1071 GP65536  LDA   #POS65536
9213: A0
Unknown label in line: 1072
                1072          LDY   #>POS65536
9216: 60        1073          RTS
                1074
                1075 * Get address of array which name is pointed to by
                1076 * TXTPTR. If no array is found, then the called
                1077 * ROM routine would have created one so we´ll have
                1078 * to rollback such creation and exit.
                1079 NGTA2     DO    KOPT16
9217: D4 6D     1080          PEI   STREND
9219: 20 8A 7E  1087          JSR   NGETARPT
921C: FA        1088          PLX
921D: 68        1089          PLA
```

```
Bad branch in line: 1090
                1090          BCS   :1          found existing array
9220: 85 6E     1091          STA   STREND+1    Do the rollback
9222: 86 6D     1092          STX   STREND
                1093 :1       DO    KOPT-K65C02
9224: 64 14     1097          STZ   SUBFLG
9226: 60        1099          RTS
                1100
                1101          PUT   PEERFORNEXT
               >1     * Module en charge du traitement de boucles FOR/NEXT
               >2     * en variante classique comme en variante FOREACH
               >3     GTFORPNT EQU   $D365
               >4     GETSPA   EQU   $E452      Get mem. space for new string
               >5
9227: 4C 76 DD >6     ]ERR     JMP   GOTMIERR
922A: 20 F2 8A >7     FEFOR    JSR   NCHKCOM    Ensure trailing comma
922D: D4 85    >9              PEI   FORPNT
922F: D4 11    >10             PEI   VALTYP


--End assembly, 8012 bytes, Errors: 89


Symbol table - alphabetical order:

   A1L     =$3C        A2L     =$3E        A4L     =$42        ABSOL8  =$7D9E
   ABSOLUTE=$7E5F   ?  ACTR    =$9B        ADB1    =$42C8      ADB2    =$42E0
   ADDON   =$D998      ADRUSR  =$01        ADT1    =$42D4      ADT2    =$42EC
   ALKCACH =$81FF      ALTZP   =$C009      ARET    =$7E50   ?  ARG     =$A5
   AROMBA  =$47CF      ARYPNT  =$94        ARYTAB  =$6B        ARYVAR  =$D033
   AUXBANK =$BF        AUXPTR  =$06        AXARTAB =$D079   ?  AXARYPNT=$D079
?  AXARYPT2=$D07E      AXHIMEM =$BF00   ?  AXOFFSET=$D07B      AXVALUE =$D07E
?  AYINT   =$E10C      BADNAM  =$7EB9      BAMBS   =$0778      BANCLD  =$8689
   BAXHI   =$0578      BAXLO   =$0478      BAYHI   =$05F8      BAYLO   =$04F8
   BIGRECON=$42F8      BISVTYP =$BE        BTMEL   =$D139      CALLFUNC=$8ADB
   CFA     =$4392      CFM     =$438E      CGARBAG =$7AA6      CH      =$24
   CHKMEM  =$D3D6      CHKNUM  =$DD6A      CHKSTR  =$DD6C      CMPCLAMP=$8EC0
   CNVT1   =$81F7      CODE1BF =$453C      CODE1GC =$4696      CODE1GCF=$47CF
   CODE1LC =$45F8      CODE2BF =$45F8      CODE2LC =$467B      COLLECTR=$8576
   COMBYTE =$E74C      COMCLAMP=$8EE3      COMCLEAR=$8F18      COMCMPLX=$8AD3
   COMINT1 =$912C      COMINT2 =$917C      COMINT4 =$90A3      COMLBS  =$8F97
   COMLET2 =$83B2      COMLISO =$8889      COMMON  =$9061      COMMON9 =$905C
   COMMONG =$848F      COMPOFST=$830D      COMPOS  =$8F4B      COMREAD =$8F1D
   COMREST =$8568      COMRST  =$7ACC      COMRSTC =$7AD4      COMWAIT =$9280
   COMX1   =$837A      CONINT  =$E6FB      CONV1628=$7E72      COPYROM =$4406
   CRDO    =$DAFB      CURLIN  =$75        CURLSV  =$F6        DATA    =$D995
   DATA1IDX=$4396      DATA1VAL=$439C   ?  DATAN   =$D9A3      DBUFP   =$9D00
   DEBUTGET=$7AA6      DEBUTGOT=$7AED      DECTPTR =$865E      DEFFLG  =$C1
?  DEST    =$60        DIMFLG  =$10        DINSIRQV=$8EA6      DIVEND  =$C2
   DIVSOR  =$C0        DSCLEN  =$8F        DSCTMP  =$9D        DVAR    =$D08B
   DVARS   =$D07C      DVARTS  =$D0DF      DVZERROR=$7DEF      E06     =$8701
   EK      =$41C9   ?  ELMSIZ  =$D07D   MD EMOV    =$8000      ENDCHR  =$0E
   ENDRNG  =$876E   ?  ERRDIR  =$E306      ERRFLG  =$D8     ?  ERRLIN  =$DA
?  ERRNUM  =$DE     ?  ERRPOS  =$DC     ?  ERRSTK  =$DF        ERR_BSCR=$6B
   ERR_RDIM=$78        ERR_SYNT=$10        EXPLIC? =$7EC4      FAC     =$9D
   FACLO   =$A1        FACMO   =$A0        FACSIGN =$A2        FADD    =$E7BE
   FAE2    =$81A2      FAE3    =$81A3      FC      =$3C        FCOMP   =$EBB2
```

```
   FCSTK   =$40     ?  FDIV    =$EA66      FE      =$3D     ?  FEFOR   =$92E1
   FINMOUSE=$8F1A      FM      =$3E        FMSTK   =$42     ?  FMULT   =$E97F
   FNDLIN  =$D61A   ?  FNDVAR  =$D006      FNDVAR2 =$7AA6      FNDVARX2=$D00E
   FORPNT  =$85     ?  FREESPC =$71        FREFAC  =$E600      FRETOP  =$6F
?  FRMELM  =$89A4      FRMELMLP=$89A1      FRMEVL  =$DD7B      FRMNUM  =$DD67
?  FRMSTCK3=$DE20      FRSTIM  =$832C   ?  FSUB    =$E7A7      FX      =$3F
   FXSTK   =$41        G81     =$BFAB      G83     =$BFA4      GARBAG  =$E484
   GDVARTS =$D086      GETADR  =$E752      GETARY  =$E0ED      GETARY2 =$E0EF
   GETBYT  =$E6F8   ?  GETSPA  =$E452      GFLAG   =$C0        GGO2TMER=$8BAE
   GIQERR2 =$824F      GIVAYF  =$E2F2      GME     =$8172      GN32768 =$92BA
   GN65536 =$92C4      GNARRAY =$801E      GNPTRGET=$7AEA      GODVZERR=$EAE1
   GOIQ    =$892B      GOIQERR =$E199      GOOVFERR=$E8D5      GOSTLERR=$E5B2
?  GOSVCUR =$8328      GOSYNERR=$82E1      GOTMIERR=$DD76   MD GOTO    =$8000
   GP32768 =$92BF      GP65536 =$92C9      GRBPAS  =$D0EC      GSE     =$819F
   GSNERR2 =$824C      GSNERR3 =$863A   ?  GTFORPNT=$D365     GTLT    =$7EAA
?  GTMERR2 =$8252      GZAUXRT =$BF00      HE2E8   =$8B02      HIMEM   =$73
   HNDLEADR=$839F      HNDLEBC =$7CB5      HNDLEIC =$7C5B      HNDLEINT=$7BC8
   HNDLEIX =$7C56   ?  HNDLEIY =$7BF1   ?  HNDLEREA=$7B80      HNDLESTR=$7B9D
?  HNDLSBAD=$7C6C   ?  HNDLSBDV=$7C9F   ?  HNDLSBMI=$7C79   ?  HNDLSBMU=$7C81
?  HNDLSIAD=$7C1D   ?  HNDLSIDV=$7C46   ?  HNDLSIMI=$7C28   ?  HNDLSIMU=$7C47
?  HNDLUBAD=$7C65   ?  HNDLUBDV=$7C9E   ?  HNDLUBMI=$7C72   ?  HNDLUBMU=$7C80
?  HNDLUIAD=$7C05   ?  HNDLUIDV=$7C34   ?  HNDLUIMI=$7C11   ?  HNDLUIMU=$7C35
   HNOK    =$9249      IDMOCL  =$BD        IDX0    =$C0        INDEX   =$5E
   INITBF  =$44C5   ?  INITLC  =$467B      INSDS2  =$F88C      INSIRQV =$8E85
   INTTYP  =$12        INTTYPSV=$C7        IRQHDLR =$8E33      IRQTBLE =$BFB2
   IRQV    =$03FE      ISAUXMEM=$8255      ISCNTC  =$D858      ISHOSTOK=$9239
   ISLETC  =$E07D      ISMOUSH =$9241      IVALARG =$8ED9      K6502   =$00
   K65816  =$01        K65C02  =$01     ?  KANCACH =$04        KILLEMAL=$8D49
   KNEW    =$01     ?  KNEW2   =$01        KOPT    =$01        KOPT16  =$01
   KOPTLNG32=$01       KOPTLNG33=$00     ?  KSNCACH =$04          KWELMSIZ=$81C8
   KX3     =$8D44      L08     =$87B3      L088    =$87B1      L3      =$8A14
   LBS00   =$82A0   ?  LBS03   =$7CD1      LBS04   =$8C23      LBS041  =$8CD2
   LBS06   =$8E16   ?  LBS061  =$8E18      LBS10   =$8FE1      LBS49   =$7CB8
   LBS80   =$8B14      LBS81   =$8B11      LENGTH  =$2F        LENREC  =$15
   LENTHS  =$D149      LET2    =$DA63      LETINF  =$C0        LEVELPAR=$BD
   LGSYNERR=$897A      LINGET  =$DA0C      LINNUM  =$50        LISTED  =$87F3
   LLOOP   =$7AC6      LN      =$4208   M? LOOP    =$4000      LOWTR   =$9B
   LST1LIN =$879F      LSTD?   =$879D      LTOKEN  =$8899      MACMAT  =$4378
   MAINLIST=$8776      MC      =$41E9      MC1     =$41FF      MCAND   =$C0
   MCODE   =$4380      MEMERR  =$D410      MFIN    =$8B26      MINSDS2 =$4227
   MISLETC =$82D4      MKNARRAY=$8091      MKNV    =$E09C      MODDAT  =$BF
   MODREM  =$BE        MOUSEDET=$43A2      MOVE    =$FE2C      MOVFA   =$EB53
   MOVFM   =$EAF9      MOVINS  =$E5D4   MD?MOVM    =$8000      MOVMF   =$EB2B
MD MPHX    =$8000   MD MPHY    =$8000      MPLIER  =$C2     MD MPLX    =$8000
MD MPLY    =$8000      MTFUNC  =$90D6   MD MTSB    =$8000      MULTPLSS=$E2AD
   MULTPLY1=$E2B6      MZRTAUX =$41DA      NAMFOUND=$7F52      NAMNTFND=$7F2D
   NARRAY  =$7F9D      NARRGL91=$7FE2      NCHKCLS =$8AEF      NCHKCOM =$8AF2
   NCHKOPN =$8AF5      NCR     =$87DF      NEG8    =$7DA2      NEGATE  =$7E63
   NEGOP   =$EED0      NERRH   =$9250   ?  NERRHP  =$924B      NEVAL   =$8F8C
   NEVALC  =$8F83   ?  NEWAYINT=$7CDB      NEWGARBG=$E484      NEWSTT  =$D7D2
   NEWY    =$47        NEXTC2  =$8DDE      NEXTCTX =$8DC5      NFAEP   =$8175
   NFRMEVL =$8AFA      NFRMNUM =$898C      NGARBAG =$823C      NGETARPT=$7E8A
   NGETBYT =$8B1C      NGTA2   =$92CE      NILLM   =$924E   ?  NLET2   =$7B90
   NMAKINT =$81DB      NMOVINS =$7BC1   ?  NOPER   =$04     ?  NOUVIN  =$86B7
   NPARCHK =$8AE9      NPTRG   =$8F67      NPTRGET =$7E90      NPTRGET1=$7E96
   NPTRGETX=$8CE9      NPTRGL90=$7EFA      NPTRGTX =$7E8E      NREASON =$7F8B
   NRET    =$7E4E      NROUT   =$7CD6      NSYNCHR =$82DA      NSYNCHR2=$82DA
```

```
   NUMDIM  =$0F        NUMELS  =$08        NUMELS2 =$10        NWGVAYF =$8B0C
   NXLST   =$8782      NZTAB   =$D127      OFFSET  =$C2        OFFX16  =$42BF
   OKP1GET =$7ADE      OLDTEXT =$79        OLDTPTR =$79        OPBASE  =$425E
   OPCLC   =$425E      OPLDXYI =$42B6      OPPHP   =$4275      OPPLP   =$4282
   OPREP   =$4296      OPRND   =$44        OPSEC   =$4261      OPSEP   =$4293
   OPXCE   =$4266      OUTDO   =$DB5C      OUTSPC  =$DB57      PARTIAL =$BE
   PCADJ   =$F953      PCL     =$3A        PTR2    =$1C        QINT    =$EBF2
   R       =$8639      R0      =$8D5B   ?  RAZPF   =$82E4      RCLEAR  =$82CE
   RCLM    =$05        RCLMAUX =$8296   ?  RCLR    =$03        RD80STOR=$C018
?  RDEF    =$85CD      RDEFSUB =$8634      RDEFUSR =$84B3      RDIM    =$88E6
   RDIMERR =$808C      RDLCBNK2=$C011      RDLCRAM =$C012      REASON  =$D3E3
   RECON   =$8719      RECON1  =$8715   ?  RECON2  =$871D      REMSTK  =$F8
   RESTOR  =$8D86      RESTOR1 =$8D65      RESTOR2 =$8D6F      RESTORC =$8DA8
   RESTORD =$8D5F      RESTORF =$8DA7   ?  RESTORX =$8D98      RESULT  =$62
   RET1    =$7C04      RET3    =$8A11      RETOUR  =$8548      RETOURM =$91B4
   RETOURT =$91B7      RETURN  =$874B      RFFVL   =$89E9      RHOM    =$06
   RIIF    =$897D      RINI    =$07     ?  RLET    =$7B0E      RLET1   =$7B35
?  RMTCTRL =$8D1F   ?  RNEW    =$82C8   ?  RNEWINST=$91C8      RNEWISUI=$8D1A
   RNI2    =$91E9   ?  RONERR  =$88CB      ROUT0   =$8B74      ROUT10  =$8EEC
   ROUT11  =$8B27   ?  ROUT1X  =$8641      ROUT1Y  =$863D      ROUT4   =$8BB1
   ROUTGEN =$8B4C      RPOS    =$04        RREAD   =$02     ?  RRETURN =$88C1
?  RRUN    =$82BF      RSETM   =$00        RSRVM   =$01        RST100  =$7AC4
   RST101  =$7AC6      RST102  =$7ACC      RST103  =$7ACC      RSTALTM =$85A9
   RSTCURRM=$859E      RUSR    =$83C6      RVRAI   =$892E      RW2     =$92A8
?  RWAIT   =$926F      SAVALTM =$85BF      SAVCURRM=$85B4      SAVER   =$8DEC
   SAVERC  =$8E23      SCDCH2  =$7EBC      SCNDTIM =$8384      SCTR    =$9B
   SDIV    =$7DF2      SDIV8   =$7D4B      SENDCHR =$87CF      SETINITX=$82FF
   SETITS  =$7C60      SETLTR  =$8DBC      SETUPB  =$8667      SETUPD  =$867E
   SETVYA  =$E0DE      SKIPC   =$8936      SLKCACH =$7F57   MD?SMOVE   =$8000
   SMUL    =$7DAD      SMUL8   =$7D13   ?  SNERR   =$8089      SNGFLT  =$E301
?  STACK   =$0100   MD?STD     =$8000   ?  STDLIS  =$874C      STDZP   =$C008
MD STID    =$8000      STREND  =$6D        STRING1 =$AB        STRNG   =$19
   STRNG1  =$AC        STRNG2  =$AD        STRSPA  =$E3DD      STRTRNG =$8758
   SUBERR  =$E196      SUBFLG  =$14        SUBSERR =$8086   ?  SUITE   =$4000
   SVARS   =$D022      SWPIO   =$8E07   ?  SWREINIT=$90A6      SYNERR  =$DEC9
?  TELMS   =$D073      TEST2E  =$4442      TFUNC   =$9114      TIMEINST=$9076
   TMERR   =$DD76      TOKADD  =$C8        TOKCHRD =$E7        TOKDATA =$83
?  TOKDEF  =$B8        TOKDIM  =$86     ?  TOKDIV  =$CB        TOKEN?  =$87FC
   TOKEQUAL=$D0     ?  TOKFN   =$C2     ?  TOKFOR  =$81        TOKFRE  =$D6
   TOKGOSUB=$B0        TOKGOTO =$AB     ?  TOKIF   =$AD     ?  TOKINT  =$D3
   TOKMINUS=$C9     ?  TOKMUL  =$CA     ?  TOKNOT  =$C6        TOKREM  =$B2
?  TOKSCRN =$D7        TOKSGN  =$D2     ?  TOKSTEP =$C7        TOKSTRD =$E4
   TOKTABL =$D0D0   ?  TOKTO   =$C1        TOKUSR  =$D5        TOMOUSE =$90BD
   TRACE   =$D805      TRCFLG  =$F2        TXTPSV  =$F4        TXTPTR  =$B8
   TYPMOD  =$C1        ULERR   =$D97C      USDIV   =$7E1D      USDIV8  =$7D78
   USEOLDAR=$8023      USMUL   =$7DCB      USMUL8  =$7D30   ?  USRMOD  =$00
   V3      =$8479      VALTYP  =$11        VALTYPSV=$C8        VARNAM  =$81
   VARPNT  =$83        VARPT   =$D159      VARTAB  =$69        VECTUSR =$0A
?  VECZAUX =$03ED      VENT1IT =$0C        VENT1NAM=$09     ?  VENT1PTR=$0D
?  VENT1VT =$0B        VENT2IT =$12        VENT2NAM=$0F     ?  VENT2PTR=$13
?  VENT2VT =$11        VERSION =$15        VLET    =$DA46      VLINPRT =$87F9
   VPNT    =$A0        VPTRGET =$DFEF      VSRTIT  =$06        VSRTNAM =$03
   VSRTPTR =$07     ?  VSRTVT  =$05        XFER    =$C314   ?  XFRMMOT1=$8650
   XFROMMOT=$8653      XMFIN   =$8BF7      XMFIN1  =$8C20      XMFIN2  =$8C1D
   XSAV    =$B4        XSUITE  =$8A0F      XXSAV   =$1B        YSAV    =$B5
   ZAUXB   =$D018      ZAUXOFFT=$BFB8      ZAUXRET =$BF3E      ZAUXRT  =$D013
   ZAUXRT0 =$D018      ZAUXRT1 =$D02D      ZAUXRT2 =$D035      ZAUXRT3 =$D000
```

```
   ZCOMRT12=$D06B      ZEROPRT =$7E52      ZGCP2   =$BF93      ZGCPARMS=$BF7E
   ZNG     =$BF9E      ZPRT8   =$7D93      ZRTAUX  =$822F   V? ]ERR    =$9227
V  ]ERR1   =$8F48   V  ]ERRS   =$7C66   V  ]JLOOP  =$8F4E   V  ]LOOP   =$91EE
V  ]LOOP1  =$7B88   V  ]RET    =$918B
```

Symbol table - numerical order:

```
   K6502   =$00        KOPTLNG33=$00     ?  USRMOD  =$00        RSETM   =$00
   K65C02  =$01        K65816  =$01        KOPT    =$01        KNEW    =$01
?  KNEW2   =$01        KOPTLNG32=$01        KOPT16  =$01        ADRUSR  =$01
   RSRVM   =$01        RREAD   =$02        VSRTNAM =$03     ?  RCLR    =$03
?  KSNCACH =$04     ?  KANCACH =$04     ?  NOPER   =$04        RPOS    =$04
?  VSRTVT  =$05        RCLM    =$05        AUXPTR  =$06        VSRTIT  =$06
   RHOM    =$06        VSRTPTR =$07        RINI    =$07        NUMELS  =$08
   VENT1NAM=$09        VECTUSR =$0A     ?  VENT1VT =$0B        VENT1IT =$0C
?  VENT1PTR=$0D        ENDCHR  =$0E        NUMDIM  =$0F        VENT2NAM=$0F
   DIMFLG  =$10        NUMELS2 =$10        ERR_SYNT=$10        VALTYP  =$11
?  VENT2VT =$11        INTTYP  =$12        VENT2IT =$12     ?  VENT2PTR=$13
   SUBFLG  =$14        VERSION =$15        LENREC  =$15        STRNG   =$19
   XXSAV   =$1B        PTR2    =$1C        CH      =$24        LENGTH  =$2F
   PCL     =$3A        A1L     =$3C        FC      =$3C        FE      =$3D
   A2L     =$3E        FM      =$3E        FX      =$3F        FCSTK   =$40
   FXSTK   =$41        A4L     =$42        FMSTK   =$42        OPRND   =$44
   NEWY    =$47        LINNUM  =$50        INDEX   =$5E     ?  DEST    =$60
   RESULT  =$62        VARTAB  =$69        ARYTAB  =$6B        ERR_BSCR=$6B
   STREND  =$6D        FRETOP  =$6F     ?  FREESPC =$71        HIMEM   =$73
   CURLIN  =$75        ERR_RDIM=$78        OLDTPTR =$79        OLDTEXT =$79
?  TOKFOR  =$81        VARNAM  =$81        TOKDATA =$83        VARPNT  =$83
   FORPNT  =$85        TOKDIM  =$86        DSCLEN  =$8F        ARYPNT  =$94
   LOWTR   =$9B        SCTR    =$9B     ?  ACTR    =$9B        FAC     =$9D
   DSCTMP  =$9D        FACMO   =$A0        VPNT    =$A0        FACLO   =$A1
   FACSIGN =$A2     ?  ARG     =$A5        TOKGOTO =$AB        STRING1 =$AB
   STRNG1  =$AC     ?  TOKIF   =$AD        STRNG2  =$AD        TOKGOSUB=$B0
   TOKREM  =$B2        XSAV    =$B4        YSAV    =$B5     ?  TOKDEF  =$B8
   TXTPTR  =$B8        IDMOCL  =$BD        LEVELPAR=$BD        PARTIAL =$BE
   MODREM  =$BE        BISVTYP =$BE        AUXBANK =$BF        MODDAT  =$BF
   MCAND   =$C0        DIVSOR  =$C0        LETINF  =$C0        GFLAG   =$C0
   IDX0    =$C0     ?  TOKTO   =$C1        TYPMOD  =$C1        DEFFLG  =$C1
?  TOKFN   =$C2        MPLIER  =$C2        DIVEND  =$C2        OFFSET  =$C2
?  TOKNOT  =$C6     ?  TOKSTEP =$C7        INTTYPSV=$C7        TOKADD  =$C8
   VALTYPSV=$C8        TOKMINUS=$C9     ?  TOKMUL  =$CA     ?  TOKDIV  =$CB
   TOKEQUAL=$D0        TOKSGN  =$D2     ?  TOKINT  =$D3        TOKUSR  =$D5
   TOKFRE  =$D6     ?  TOKSCRN =$D7        ERRFLG  =$D8     ?  ERRLIN  =$DA
?  ERRPOS  =$DC     ?  ERRNUM  =$DE     ?  ERRSTK  =$DF        TOKSTRD =$E4
   TOKCHRD =$E7        TRCFLG  =$F2        TXTPSV  =$F4        CURLSV  =$F6
   REMSTK  =$F8     ?  STACK   =$0100   ?  VECZAUX =$03ED      IRQV    =$03FE
   BAXLO   =$0478      BAYLO   =$04F8      BAXHI   =$0578      BAYHI   =$05F8
   BAMBS   =$0778   MD EMOV    =$8000   MD?STD     =$8000   MD STID    =$8000
MD?MOVM    =$8000   MD?SMOVE   =$8000   M? LOOP    =$4000   MD MPHX    =$8000
MD MPHY    =$8000   MD MPLX    =$8000   MD MPLY    =$8000   MD MTSB    =$8000
MD GOTO    =$8000   ?  SUITE   =$4000      EK      =$41C9      MZRTAUX =$41DA
   MC      =$41E9      MC1     =$41FF      LN      =$4208      MINSDS2 =$4227
   OPBASE  =$425E      OPCLC   =$425E      OPSEC   =$4261      OPXCE   =$4266
   OPPHP   =$4275      OPPLP   =$4282      OPSEP   =$4293      OPREP   =$4296
   OPLDXYI =$42B6      OFFX16  =$42BF      ADB1    =$42C8      ADT1    =$42D4
   ADB2    =$42E0      ADT2    =$42EC      BIGRECON=$42F8      MACMAT  =$4378
```

```
   MCODE   =$4380      CFM     =$438E      CFA     =$4392      DATA1IDX=$4396
   DATA1VAL=$439C      MOUSEDET=$43A2      COPYROM =$4406      TEST2E  =$4442
   INITBF  =$44C5      CODE1BF =$453C      CODE2BF =$45F8      CODE1LC =$45F8
   CODE2LC =$467B   ?  INITLC  =$467B      CODE1GC =$4696      CODE1GCF=$47CF
   AROMBA  =$47CF      FNDVAR2 =$7AA6      CGARBAG =$7AA6      DEBUTGET=$7AA6
   RST100  =$7AC4      RST101  =$7AC6      LLOOP   =$7AC6      RST102  =$7ACC
   RST103  =$7ACC      COMRST  =$7ACC      COMRSTC =$7AD4      OKP1GET =$7ADE
   GNPTRGET=$7AEA      DEBUTGOT=$7AED   ?  RLET    =$7B0E      RLET1   =$7B35
?  HNDLEREA=$7B80   V  ]LOOP1  =$7B88   ?  NLET2   =$7B90      HNDLESTR=$7B9D
   NMOVINS =$7BC1      HNDLEINT=$7BC8   ?  HNDLEIY =$7BF1      RET1    =$7C04
?  HNDLUIAD=$7C05   ?  HNDLUIMI=$7C11   ?  HNDLSIAD=$7C1D   ?  HNDLSIMI=$7C28
?  HNDLUIDV=$7C34   ?  HNDLUIMU=$7C35   ?  HNDLSIDV=$7C46   ?  HNDLSIMU=$7C47
   HNDLEIX =$7C56      HNDLEIC =$7C5B      SETITS  =$7C60   ?  HNDLUBAD=$7C65
V  ]ERRS   =$7C66   ?  HNDLSBAD=$7C6C   ?  HNDLUBMI=$7C72   ?  HNDLSBMI=$7C79
?  HNDLUBMU=$7C80   ?  HNDLSBMU=$7C81   ?  HNDLUBDV=$7C9E   ?  HNDLSBDV=$7C9F
   HNDLEBC =$7CB5      LBS49   =$7CB8   ?  LBS03   =$7CD1      NROUT   =$7CD6
?  NEWAYINT=$7CDB      SMUL8   =$7D13      USMUL8  =$7D30      SDIV8   =$7D4B
   USDIV8  =$7D78      ZPRT8   =$7D93      ABSOL8  =$7D9E      NEG8    =$7DA2
   SMUL    =$7DAD      USMUL   =$7DCB      DVZERROR=$7DEF      SDIV    =$7DF2
   USDIV   =$7E1D      NRET    =$7E4E      ARET    =$7E50      ZEROPRT =$7E52
   ABSOLUTE=$7E5F      NEGATE  =$7E63      CONV1628=$7E72      NGETARPT=$7E8A
   NPTRGTX =$7E8E      NPTRGET =$7E90      NPTRGET1=$7E96      GTLT    =$7EAA
   BADNAM  =$7EB9      SCDCH2  =$7EBC      EXPLIC? =$7EC4      NPTRGL90=$7EFA
   NAMNTFND=$7F2D      NAMFOUND=$7F52      SLKCACH =$7F57      NREASON =$7F8B
   NARRAY  =$7F9D      NARRGL91=$7FE2      GNARRAY =$801E      USEOLDAR=$8023
   SUBSERR =$8086   ?  SNERR   =$8089      RDIMERR =$808C      MKNARRAY=$8091
   GME     =$8172      NFAEP   =$8175      GSE     =$819F      FAE2    =$81A2
   FAE3    =$81A3      KWELMSIZ=$81C8      NMAKINT =$81DB      CNVT1   =$81F7
   ALKCACH =$81FF      ZRTAUX  =$822F      NGARBAG =$823C      GSNERR2 =$824C
   GIQERR2 =$824F   ?  GTMERR2 =$8252      ISAUXMEM=$8255      RCLMAUX =$8296
   LBS00   =$82A0   ?  RRUN    =$82BF   ?  RNEW    =$82C8      RCLEAR  =$82CE
   MISLETC =$82D4      NSYNCHR =$82DA      NSYNCHR2=$82DA      GOSYNERR=$82E1
?  RAZPF   =$82E4      SETINITX=$82FF      COMPOFST=$830D   ?  GOSVCUR =$8328
   FRSTIM  =$832C      COMX1   =$837A      SCNDTIM =$8384      HNDLEADR=$839F
   COMLET2 =$83B2      RUSR    =$83C6      V3      =$8479      COMMONG =$848F
   RDEFUSR =$84B3      RETOUR  =$8548      COMREST =$8568      COLLECTR=$8576
   RSTCURRM=$859E      RSTALTM =$85A9      SAVCURRM=$85B4      SAVALTM =$85BF
?  RDEF    =$85CD      RDEFSUB =$8634      R       =$8639      GSNERR3 =$863A
   ROUT1Y  =$863D   ?  ROUT1X  =$8641   ?  XFRMMOT1=$8650      XFROMMOT=$8653
   DECTPTR =$865E      SETUPB  =$8667      SETUPD  =$867E      BANCLD  =$8689
?  NOUVIN  =$86B7      E06     =$8701      RECON1  =$8715      RECON   =$8719
?  RECON2  =$871D      RETURN  =$874B   ?  STDLIS  =$874C      STRTRNG =$8758
   ENDRNG  =$876E      MAINLIST=$8776      NXLST   =$8782      LSTD?   =$879D
   LST1LIN =$879F      L088    =$87B1      L08     =$87B3      SENDCHR =$87CF
   NCR     =$87DF      LISTED  =$87F3      VLINPRT =$87F9      TOKEN?  =$87FC
   COMLISO =$8889      LTOKEN  =$8899   ?  RRETURN =$88C1   ?  RONERR  =$88CB
   RDIM    =$88E6      GOIQ    =$892B      RVRAI   =$892E      SKIPC   =$8936
   LGSYNERR=$897A      RIIF    =$897D      NFRMNUM =$898C      FRMELMLP=$89A1
?  FRMELM  =$89A4      RFFVL   =$89E9      XSUITE  =$8A0F      RET3    =$8A11
   L3      =$8A14      COMCMPLX=$8AD3      CALLFUNC=$8ADB      NPARCHK =$8AE9
   NCHKCLS =$8AEF      NCHKCOM =$8AF2      NCHKOPN =$8AF5      NFRMEVL =$8AFA
   HE2E8   =$8B02      NWGVAYF =$8B0C      LBS81   =$8B11      LBS80   =$8B14
   NGETBYT =$8B1C      MFIN    =$8B26      ROUT11  =$8B27      ROUTGEN =$8B4C
   ROUT0   =$8B74      GGO2TMER=$8BAE      ROUT4   =$8BB1      XMFIN   =$8BF7
   XMFIN2  =$8C1D      XMFIN1  =$8C20      LBS04   =$8C23      LBS041  =$8CD2
   NPTRGETX=$8CE9      RNEWISUI=$8D1A   ?  RMTCTRL =$8D1F      KX3     =$8D44
   KILLEMAL=$8D49      R0      =$8D5B      RESTORD =$8D5F      RESTOR1 =$8D65
```

```
   RESTOR2 =$8D6F      RESTOR  =$8D86   ?  RESTORX =$8D98      RESTORF =$8DA7
   RESTORC =$8DA8      SETLTR  =$8DBC      NEXTCTX =$8DC5      NEXTC2  =$8DDE
   SAVER   =$8DEC      SWPIO   =$8E07      LBS06   =$8E16   ?  LBS061  =$8E18
   SAVERC  =$8E23      IRQHDLR =$8E33      INSIRQV =$8E85      DINSIRQV=$8EA6
   CMPCLAMP=$8EC0      IVALARG =$8ED9      COMCLAMP=$8EE3      ROUT10  =$8EEC
   COMCLEAR=$8F18      FINMOUSE=$8F1A      COMREAD =$8F1D   V  ]ERR1   =$8F48
   COMPOS  =$8F4B   V  ]JLOOP  =$8F4E      NPTRG   =$8F67      NEVALC  =$8F83
   NEVAL   =$8F8C      COMLBS  =$8F97      LBS10   =$8FE1      COMMON9 =$905C
   COMMON  =$9061      TIMEINST=$9076      COMINT4 =$90A3   ?  SWREINIT=$90A6
   TOMOUSE =$90BD      MTFUNC  =$90D6      TFUNC   =$9114      COMINT1 =$912C
   COMINT2 =$917C   V  ]RET    =$918B      RETOURM =$91B4      RETOURT =$91B7
?  RNEWINST=$91C8      RNI2    =$91E9   V  ]LOOP   =$91EE   V? ]ERR    =$9227
   ISHOSTOK=$9239      ISMOUSH =$9241      HNOK    =$9249   ?  NERRHP  =$924B
   NILLM   =$924E      NERRH   =$9250   ?  RWAIT   =$926F      COMWAIT =$9280
   RW2     =$92A8      GN32768 =$92BA      GP32768 =$92BF      GN65536 =$92C4
   GP65536 =$92C9      NGTA2   =$92CE   ?  FEFOR   =$92E1      DBUFP   =$9D00
   AXHIMEM =$BF00      GZAUXRT =$BF00      ZAUXRET =$BF3E      ZGCPARMS=$BF7E
   ZGCP2   =$BF93      ZNG     =$BF9E      G83     =$BFA4      G81     =$BFAB
   IRQTBLE =$BFB2      ZAUXOFFT=$BFB8      STDZP   =$C008      ALTZP   =$C009
   RDLCBNK2=$C011      RDLCRAM =$C012      RD80STOR=$C018      XFER    =$C314
   ZAUXRT3 =$D000   ?  FNDVAR  =$D006      FNDVARX2=$D00E      ZAUXRT  =$D013
   ZAUXB   =$D018      ZAUXRT0 =$D018      SVARS   =$D022      ZAUXRT1 =$D02D
   ARYVAR  =$D033      ZAUXRT2 =$D035      ZCOMRT12=$D06B   ?  TELMS   =$D073
   AXARTAB =$D079   ?  AXARYPNT=$D079   ?  AXOFFSET=$D07B      DVARS   =$D07C
?  ELMSIZ  =$D07D      AXVALUE =$D07E   ?  AXARYPT2=$D07E      GDVARTS =$D086
   DVAR    =$D08B      TOKTABL =$D0D0      DVARTS  =$D0DF      GRBPAS  =$D0EC
   NZTAB   =$D127      BTMEL   =$D139      LENTHS  =$D149      VARPT   =$D159
?  GTFORPNT=$D365      CHKMEM  =$D3D6      REASON  =$D3E3      MEMERR  =$D410
   FNDLIN  =$D61A      NEWSTT  =$D7D2      TRACE   =$D805      ISCNTC  =$D858
   ULERR   =$D97C      DATA    =$D995      ADDON   =$D998   ?  DATAN   =$D9A3
   LINGET  =$DA0C      VLET    =$DA46      LET2    =$DA63      CRDO    =$DAFB
   OUTSPC  =$DB57      OUTDO   =$DB5C      FRMNUM  =$DD67      CHKNUM  =$DD6A
   CHKSTR  =$DD6C      GOTMIERR=$DD76      TMERR   =$DD76      FRMEVL  =$DD7B
?  FRMSTCK3=$DE20      SYNERR  =$DEC9      VPTRGET =$DFEF      ISLETC  =$E07D
   MKNV    =$E09C      SETVYA  =$E0DE      GETARY  =$E0ED      GETARY2 =$E0EF
?  AYINT   =$E10C      SUBERR  =$E196      GOIQERR =$E199      MULTPLSS=$E2AD
   MULTPLY1=$E2B6      GIVAYF  =$E2F2      SNGFLT  =$E301   ?  ERRDIR  =$E306
   STRSPA  =$E3DD   ?  GETSPA  =$E452      GARBAG  =$E484      NEWGARBG=$E484
   GOSTLERR=$E5B2      MOVINS  =$E5D4      FREFAC  =$E600      GETBYT  =$E6F8
   CONINT  =$E6FB      COMBYTE =$E74C      GETADR  =$E752   ?  FSUB    =$E7A7
   FADD    =$E7BE      GOOVFERR=$E8D5   ?  FMULT   =$E97F   ?  FDIV    =$EA66
   GODVZERR=$EAE1      MOVFM   =$EAF9      MOVMF   =$EB2B      MOVFA   =$EB53
   FCOMP   =$EBB2      QINT    =$EBF2      NEGOP   =$EED0      INSDS2  =$F88C
   PCADJ   =$F953      MOVE    =$FE2C
```