

Listing 1 for ProDOS BASIC Program Commands

AS.CMDS (This code already exists in the ProDOS BI. There is no need to enter it.)

```

1 .....
2 .....
3 .....
4 .....
5 ..... AS.CMDS .....
6 ..... APPLESOFT PROGRAM FILE COMMANDS .....
7 ..... ProDOS BASIC INTERPRETER version 1.1 .....
8 ..... BI 1.0 addresses bracketed .....
9 ..... Interpreted by Sandy Mossberg ..... Merlin Pro
10 .....
11 .....
12 .....
13 .....
14 * GENERAL:
15
16 CH = $24 :Cursor column
17 PROMPT = $33 :Prompt character
18 PCL = $3A :Pointer
19 AIL = $3C :Pointer
20
21 * BI PROPER
22
23 RSTINCP = $9A17 :Restore I/O intercepts
24 SAVIOTRU = $9A8D :Save true I/O handlers
25 TRACECHR = $9F61 :[$9F98] Trace character image
26 SETSTATE = $9F76 :[$9FAD] Set STATE handlers
27 PACKVAR = $A40D :[$A449] Pack variables/strings
28 UNPKVAR = $A472 :[$A4AF] Unpack vars./strings
29 SETSYNER = $A839 :[$A879] Generate SYNTAX ERROR
30 FILCREAT = $AD46 :[$AD8B] Create file
31 BRUNCMD = $AE16 :[$AE5B] BRUN command
32 GOCLOSE = $AF94 :[$AFFC] Close file
33 GOREAD = $AF98 :[$B000] Read from file
34 GOWRITE = $AF9C :[$B004] Write to file
35 OPENREIN = $B194 :[$B1EE] Open file & set ref num
36 EXECCMD = $B221 :[$B27B] EXEC command
37 CLOEXEC = $B2FB :[$B355] Close EXEC file
38 CLOSALL = $BAF2 :[$B54C] Close all nonEXEC files
39 NSINFDAT = $B7D9 :[$B833] Set file info and date
40 CHIMEM = $BC7B :HIMEM for CHAIN command
41 VRSTRADR = $BC8E :Address of variables/strings
42
43 * BI GLOBAL PAGE:
44
45 TFIPLACTV = $BE44 :Input file activity flag
46 XNUM = $BE53 :BASIC command number (0=exit)
47 FBITS = $BE56 :Found parameters
48 VADDR = $BE58 :Value for A-parameter
49 WLINE = $BE68 :Value for w-parameter
50 VTYPE = $BE6A :Value for T-parameter
51 GOSYSTEM = $BE70 :Execute MLI call via BI
52 CRAUID = $BEA5 :Auxiliary type code
53 FIACCESS = $BEB7 :Access code
54 FIFILID = $BE8B :File type code
55 FIAUXID = $BE89 :Auxiliary type code
56 SEOF = $BEC8 :End of file marker (3 bytes)
57 RMOATA = $BED7 :Buffer pointer
58
59 * SYSTEM GLOBAL PAGE
60
61 BITMAP = $BF58 :System bit map
62
63 * APPLESOFT ZERO PAGE and ROM
64
65 LINNUM = $50 :Line number
66 TXTTAB = $67 :Start of program
67 VARTAB = $69 :Bottom of simple variables
68 ARYTAB = $6B :Bottom of array variables
69 STREND = $6D :Bottom of free space
70 MEMSIZ = $73 :HIMEM
71 LOWTR = $9B :Pointer to line location
72 PRGEND = $AF :End of program space
73 TXTPTR = $B8 :Text pointer
74 ERRFLG = $D8 :ONERR flag
75 ERRNUM = $DE :ONERR code number
76 CMDLP = $D43C :Main command loop
77 FNDLIN = $D61A :Find address of line in LINNUM
78 TPTCLEAR = $D665 :Reset TXTPTR, stack, variables
79 NEWSTT = $D7D2 :Execute new statement
80 NORMAL = $F273 :Set normal text mode
81
82 * PBITS(+1)-FBITS(+1)
83
84 AD = $80 :Starting address (A-parameter)
85 LINE = $0B :Line number (s-parameter)
86 T = $04 :File type (T-parameter)
87 FN1 = $01 :Filename expected
88
89 * FILE TYPES
90
91 SYS = $FF :System
92 BAS = $FC :BASIC
93 BIN = $06 :Binary
94 TXT = $04 :Text
95
96 * MLI CALLS
97

```

```

98 SETFOF = $DB
99 GETEOF = $DD
100 .....
101 * DASH Command [$ABB3]
102 .....
103 ORG $AB43
104 .....
105 * Determine file type and execute program
106 .....
107 DASHCMD LDA FIFILID :Get file type
108 CMP #BAS :#BAS
109 BEQ RUNCMD :Run Applesoft program
110 CMP #BIN :#BIN
111 BEQ #3 :BRUN binary program
112 CMP #TXT :#TXT
113 BNE #1 :#1
114 JMP EXECCMD :EXEC text file
115 CMP #SYS :#SYS
116 BEQ #2 :BRUN system program
117 .....
118 * Set for FILE TYPE MISMATCH error:
119 .....
120 LDA #SD
121 SEC
122 RTS
123 .....
124 * Execute SYSTEM file
125 .....
126 * -> Close all open files
127 .....
128 JSR CLOSALL :NonEXEC files
129 JSR CLOEXEC1 :EXEC files
130 .....
131 * -> Set A-param and free BI pages in system bit map
132 .....
133 LDA #0
134 STA VADDR :A-param (LSB)
135 STA BITMAP+13 :Free $98-9F
136 STA BITMAP+14 :Free $A0-A7
137 STA BITMAP+15 :Free $A8-AF
138 STA BITMAP+16 :Free $B0-B7
139 LDA #1
140 STA BITMAP+17 :Free $B8-BE Protect $BF
141 LDA #520 :A-param (MSB) System program
142 STA VADDR+1 :loads at $2000
143 .....
144 * -> Set T-param to SYS file type
145 .....
146 LDA #SYS
147 STA VTYPE
148 .....
149 * -> Flag address, pathname and file type parms:
150 .....
151 LDA #AD
152 STA FBITS+1 :Set FBITS+1 bit 7
153 LDA #FN1+T
154 STA FBITS :Set FBITS bits 0,2
155 .....
156 * Go to BRUN command:
157 .....
158 JMP BRUNCMD
159 .....
160 * CHAIN Command [$ABD0]
161 .....
162 CHAINCMD JSR PACKVAR :Pack variables/strings
163 LDA MEMSIZ+1
164 STA CHIMEM :Save entry HIMEM
165 LDX VRSTRADR+1 :Set load HIMEM 1
166 DEX :page below packed
167 STX MEMSIZ+1 :variables/strings
168 JSR LOAD2 :Load program (close no files)
169 LDX CHIMEM
170 STX MEMSIZ+1 :Restore entry HIMEM
171 BCS RTS20 :Error exit (from load)
172 JSR TPTCLEAR :Reset TXTPTR, stack, variable
173 JSR UNPKVAR :Unpack variables/strings
174 LDA #0 :Clear ONERR flag
175 BEQ RUN1 :Always
176 .....
177 * RUN Command [$ABF2]
178 .....
179 * Check BI vs Applesoft RUN command
180 .....
181 RUNCMD LDA #0
182 STA TFIPLACTV :Clear file input flag and
183 STA ERRNUM :zero Applesoft error code
184 LDA FBITS :Applesoft RUN command has
185 LSR :no trailing filename
186 BCC RUNINIT :Filename not given
187 .....
188 * BI RUN command given so load and execute program:
189 .....
190 JSR LOAD1 :Load Applesoft file
191 BCS RTS20 :Report error
192 JSR TPTCLEAR :Reset TXTPTR, stack, variables
193 STA ERRFLG :Clear ONERR flag
194 JSR SETLINUM :Start at correct line number
195 JSR SAVIOTRU :Restore BI intercepts
196 JSR RUNINIT :Initialize functions
197 JMP NEWSTT :Execute loaded program
198 .....
199 * Initialize various functions
200 .....
201 RUNINIT JSR NORMAL :Set normal text mode
202 LDA #"" :Reset normal
203 STA TRACECHR :trace char
204 LDA #=-1 :Bypass STATE 0 Input Handler
205 STA XNUM :error generation so Applesoft
206 STA PROMPT :can interpret RUN command
207 LDX #4 :Set state for
208 JSR SETSTATE :deferred mode
209 JMP SETSYNER :Set for possible syntax error
210 .....
211 * LOAD Command [$AC30]
212 .....

```

```

213 * Close nonEXEC files and load Applesoft file:
214
ABEC: 20 FE AB 215 LOADCMD JSR LOAD1
ABEF: 80 23 216 BCS RTS20 :Report error
217
218 * Warmstart Procs:
219
ABF1: 20 65 D6 220 WARMDS JSR TPTCLEAR :Reset TXTPTR, stack, variables
ABF4: 20 17 9A 221 JSR RSTINCPTR :Restore BI intercepts
ABF7: A9 00 222 LDA #0
ABF9: 85 24 223 STA CH :Start at beginning of line
ABFB: 4C 3F D4 224 JMP GMDLP43 :Reenter Applesoft (skip CR)
225
226 * Close all nonEXEC files:
227
ABFE: 20 F2 B4 228 LOAD1 JSR CLOSALL
AC01: 80 11 229 BCS RTS20 :Report error
230
231 * Load Applesoft file and set pointers:
232
AC03: 20 15 AC 233 LOAD2 JSR DOLOAD
AC06: 80 0C 234 BCS RTS20 :Load the file
AC08: 84 68 235 STY ARTAB :Bottom of arrays (LSB)
AC0A: 84 69 236 STY VARTAB :Start of simple variables (LSB)
AC0C: 84 6D 237 STY STREND :Bottom of free space (LSB)
AC0E: 86 6C 238 STX ARTAB+1 :Bottom of arrays (MSB)
AC10: 85 6A 239 STX VARTAB+1 :Start of simple variables (MSB)
AC12: 85 6E 240 STX STREND+1 :Bottom of free space (MSB)
AC14: 6E 241 RTS20 RTS :Successful load (carry clear)
242
243 * Open Applesoft file:
244
NC15: A9 01 245 DOLOAD LDA #1 :Ensure read access to file
NC17: A2 FC 246 LDX #BAS : and BAS file type
NC19: 20 94 B1 247 JSR OPENREFN :Open file, set reference number
NC1C: 80 F6 248 BCS RTS20 :Report error
249
250 * Set beginning and ending addresses of Applesoft program:
251
IC1E: A9 D1 252 LDA AGETEOF
IC20: 20 70 BE 253 JSR GOSYSTEM :GET.EOF call (get file length)
IC23: 80 EF 254 BCS RTS20 :Call error
255
IC25: A5 67 255 LDA TXTTAB
IC27: 80 07 BE 256 STA RWDATA :Beginning program address (LSB)
IC2A: 60 C8 BE 257 ADC SEOF :Add program length (LSB)
IC2D: 80 18 BE 258 LDA TXTTAB+1 :Ending program address (LSB)
IC2F: A5 67 259 STA RWDATA+1 :Beginning program address (MSB)
IC32: 80 08 BE 260 ADC SEOF+1 :Add program length (MSB)
IC35: 60 C9 BE 261 LDA TXTTAB+1 :Ending program address (MSB)
IC38: 80 59 BE 262 STA VADOR+1 :Never taken
IC3B: 80 02 263 BCS #1
264
265 * Report error if program would extend above HIMEM:
266
C3D: C5 74 267 CMP MEMSIZ+1
C3F: A9 0E 268 #1 LDA #E :PROGRAM TOO LARGE error code
C41: 80 D1 269 BCS RTS20
270
271 * Read entire Applesoft file into memory:
272
273
274
275
276
277
278 * Close Applesoft file:
279
280
281
282
283
284
285 * Set end-of-program pointer:
286
55: AE 59 BE 286 LDX VADOR+1
59: AC 58 BE 287 LDY VADOR
5C: E5 80 288 STX PRGEND+1
5E: 84 AF 289 SBC PRGEND
60: 68 290 RTS
291
292 * Check if program saved from same address as just loaded:
293
51: 20 94 AF 294 LOADLINK SEC
52: A5 67 295 LDA TXTTAB :Current load address (LSB)
54: ED 09 BE 296 SBC FIAUXID :True (saved) load address (LSB)
57: 85 3C 297 STA ALL :Offset between load/save (LSB)
59: A5 68 298 LDA TXTTAB+1 :Current load address (MSB)
5B: ED 0A BE 299 SBC FIAUXID+1 :True (saved) load address (MSB)
5E: 85 3D 300 STA ALL+1 :Offset between load/save (MSB)
70: 05 3C 301 ORA ALL
72: 18 302 CLC :Assume no error/prepare for add
73: F0 45 303 BRQ RTS21 :Program loaded to true address
304
305 * Program not loaded to true address so reset link bytes:
306
307
308 * -> Point to beginning of Applesoft program:
309
7: A5 67 309 LDX TXTTAB
7: A5 68 310 LDA TXTTAB+1
311
312 * -> Point to beginning of program line:
313
9: 86 3A 314 #1 STX PCL
8: 85 3B 315 STA PCL+1
316
317 * -> Check end-of-program marker:
318
D: A0 01 319 LDY #1 :Looking for zero high
F: 81 3A 320 LDA (PCL),Y : order link byte
1: 80 321 DEY
2: 11 3A 322 ORA (PCL),Y
4: F0 34 323 BRQ RTS21 :End of program found
324
325 * -> Adjust link bytes:
326

```

```

AC95: 81 3A 327 LGA (PCL),Y :Get link byte (LSB)
AC98: 65 3C 328 ADC AIL :Add offset (LSB)
AC9A: AA 329 TAX :Point to next link (LSB)
AC9B: 91 3A 330 STA (PCL),Y :Reset link byte (LSB)
AC9D: C8 331 INY
AC9E: 81 3A 332 LDA (PCL),Y :Get link byte (MSB)
AC9F: 65 3D 333 ADC AIL+1 :Add offset (MSB)
AC99: 91 3A 334 STA (PCL),Y :Reset link byte (MSB)
AC9A: 18 335 CLC :Clear carry for next addition
AC95: 90 E2 336 BCC #1 :Always
337
338 * Set Program Execution at Specified Line Number (SADCC)
339
340 * Check for q-parameter:
341
AC97: AD 57 BE 342 SETLINUM LDA FBITS+1
AC9A: 29 08 343 AND #LINE :Isolate FBITS+1 bit 7
AC9C: 18 344 CLC
AC9D: F0 1B 345 BRQ RTS21 :Line number not specified
346
347 * Find address of designated (or next higher) line:
348
AC9F: AD 68 BE 349 LDA VLINE
ACA2: 85 90 350 STA LENNUM
ACA4: AD 69 BE 351 LDA VLINE+1
ACA7: 85 51 352 STA LENNUM+1
ACA9: 20 1A D6 353 JSR FNOLIN :Point LOWTRK at line in LENNUM
354
355 * Point text pointer at line address minus one:
356
ACAC: 18 357 CLC
ACAD: A5 98 358 LDA LOWTR
ACAF: 69 FF 359 ADC #-1
ACB1: 85 88 360 STA TXTPTR
ACB3: A5 9C 361 LDA LOWTR+1
ACB5: 69 FF 362 ADC #-1
ACB7: 85 89 363 STA TXTPTR+1
ACB9: 18 364 CLC
ACBA: 60 365 RTS21 RTS
366
367 * SAVE Command (SAD00):
368
369 * Create Applesoft file if one does not exist:
370
ACB8: 90 22 371 SAVECMD BCC #1 :Applesoft file already exists
ACBD: A9 FC 372 LDA #BAS
ACBF: 80 6A BE 373 STA VTYPE :Put BAS file type code in PVALS
ACC2: 80 88 BE 374 STA FIFLID :Table and FILE.INFO parmlist
ACC5: A9 C3 375 LDA #FC :Allow free
ACC7: 80 87 BE 376 STA FACCESS :file access
ACCA: A5 67 377 LDA TXTTAB :in BAS type file, auxiliary
ACCC: 80 A5 BE 378 STA CRAUXID :code bytes hold default
ACCF: 80 09 BE 379 STA FIAUXID :load address of file
ACD2: A5 68 380 LDA TXTTAB+1
ACD4: 80 A5 BE 381 STA CRAUXID+1
ACD7: 80 BA BE 382 STA FIAUXID+1
ACDA: 20 46 AD 383 JSR FLICREAT :Create the BAS file
ACDD: 80 49 384 BCS RTS22 :Report error
385
386 * Open Applesoft file:
387
ACDF: A9 02 388 #1 LDA #2 :Ensure write access to file
ACE1: A2 FC 389 LDX #BAS : and BAS file type
ACE3: 20 94 B1 390 JSR OPENREFN :Open file, set reference number
ACE6: 80 40 391 BCS RTS22 :Report error
392
393 * Calculate size of Applesoft file and set EOF marker:
394
ACE8: A5 AF 395 LDA PRGEND
ACEA: 38 396 SEC
ACEB: E5 67 397 SBC TXTTAB
ACED: AA 398 TAX :File size (LSB)
ACEE: 80 C8 BE 399 STA SEOF
ACF1: A5 89 400 LDA PRGEND+1
ACF3: E5 68 401 SBC TXTTAB+1
ACF5: A8 402 TAX :File size (MSB)
ACF6: 80 C9 BE 403 STA SEOF+1
ACF9: A9 00 404 LDA #0
ACFB: 80 CA BE 405 STA SEOF+2 :File never larger than 64K
406
407 * Write Applesoft file to disk:
408
ACFE: A5 67 409 LDA TXTTAB :Start of program
AD00: 8D D7 BE 410 STA RWDATA : equals start of
AD03: A5 68 411 LDA TXTTAB+1 : data buffer for
AD05: 8D D8 BE 412 STA RWDATA+1 : WRITE call
AD08: 20 9C AF 413 JSR GWRITE :Write data
AD0B: 80 18 414 BCS RTSEOF :Report error
AD0D: A9 D0 415 LDA #SETEOF
AD12: 80 14 417 BCS GOSYSTEM :SET.EOF call
AD14: 20 94 AF 418 JSR RTS22 :Call error
AD17: 80 0F 419 JSR GOCLOSE :Close Applesoft file
420 BCS RTS22 :Report error
421
422 * Update file information if necessary:
423
AD19: A5 68 423 LDA TXTTAB+1 :Ensure that auxiliary
AD1B: A6 67 424 LDX TXTTAB : type field contains
AD1D: C0 BA BE 425 CMP FIAUXID+1 : start of program
AD20: D8 07 426 BNE SAVEINFO : (i.e. load address)
AD22: EC 09 BE 427 CPX FIAUXID
AD25: 18 428 CLC
AD26: D8 01 429 BNE SAVEINFO :Signal successful SAVE
AD28: 60 430 RTS22 RTS
431
432 * Update Applesoft file information:
433
AD29: 8E 09 BE 434 SAVEINFO STX FIAUXID :Reset auxiliary field to
AD2C: 8D BA BE 435 STA FIAUXID+1 : load address of program
AD2F: 4C 09 B7 436 JMP #NSINFDAT : and SET.FILE.INFO
437
END OF LISTING 1

```