



## TECHNICAL INFORMATION

The source is placed at START OF SOURCE when loaded, regardless of its original address.

The important pointers are:

START OF SOURCE	in	\$A,\$B	(set to \$901 unless changed)
HIMEM	in	\$C,\$D	(defaults to \$9853 in DOS 3.3, defaults to \$AA00 in ProDOS)
END OF SOURCE	in	\$E,\$F	

Note that HIMEM does not change unless a USER routine or utility program changes locations \$73, \$74. Such a change will be copied automatically into locations \$C, \$D.

### GENERAL INFORMATION (DOS 3.3 ONLY)

When you exit to BASIC or to the Monitor, these pointers are saved on the RAM card at \$E00A-\$E00F. They are restored upon re-entry to Merlin 8/16.

Entry into Merlin 8/16 replaces the current I/O hooks with the standard ones and reconnects DOS. This is the same as typing PR#0 and IN#0 from the keyboard. Entry to the Editor disconnects DOS, so that you can use labels such as INIT without disastrous consequences. Re-entry to the Main Menu disconnects any I/O hooks that you may have established via the editor's PR# command, and reconnects DOS. Exit from assembly due to completion of assembly or Control-C also disconnects I/O hooks.

Re-entry after exit to BASIC is made by the ASSEM command. Simply use ASSEM wherever a DOS command is valid, for example, at the BASIC prompt. A BRUN MERLIN or a disk boot will also provide a warm re-entry and will not reload Merlin 8 if it is already there. A reload may be forced by typing BRUN BOOT ASM which would then be a cold entry, erasing any file in memory.

The DOS 3.3 version does not perform the same volume checking as the ProDOS version. However, it is possible to simulate this with the following code:

```
LST
XXX KBD "INSERT MYFILE DISK AND TYPE 0 <RETURN>"
PAUSE
```

The assembler will stop at KBD on the first pass and assign a 0 value to XXX where XXX is any dummy label you desire. PAUSE will force a pause on the second pass and LST makes sure you will see the KBD line. On the second pass, assembly resumes when you press any key. It is not necessary to type 0 and press Return.

## GENERAL INFORMATION (ProDOS AND DOS 3.3)

In Merlin 8, if during assembly the object code exceeds usable RAM then the code will not be written to memory, but assembly will appear to proceed as normal and its output sent to the screen or printer. The only clue that this has happened, if not intentional, is that the Save Object Code command at Main Menu is disabled in this event. There is ordinarily a 16K space for object code, which can be changed with the OBJ opcode. In Merlin 16, an object code overflow generates an error message.

## SYMBOL TABLE

The symbol table is printed after assembly unless LST OFF has been used. It is displayed first sorted alphabetically and then sorted numerically. The symbol table can be canceled at any time by pressing Control-C. Stopping it in this manner will have no ill effect on the object code which was generated. The symbol table is flagged as follows:

MD = Macro Definition  
 M = Label defined within a Macro  
 V = Variable (symbols starting with "[")  
 ? = A symbol that was defined but never referenced  
 X = External symbol  
 E = Entry symbol

Local labels are not shown in the symbol table listing. In Merlin 16, this can be enabled by changing the PARMs file.

When in EDIT mode, Merlin 8/16 takes total control of input and output. The effect of typing a control character will be as described in this manual and *not* as described in the manual for your 80 column card. For example, Control-L will not blank the screen, but is the case toggle. Control-A, which acts as a case toggle on many 80 column cards, will not do this in the Editor and simply produces a Control-A in the line being edited.

## ULTRATERM INFORMATION

The Ultraterm is an 80 column display card manufactured by Videx. If you do not have this card, skip to the next page. When in the Editor, the Ultraterm mode can be altered by the ESCAPE sequence given in the Ultraterm manual. Thus, the following commands give the indicated effects:

ESC 0	..... 40 x 24	same effect as VID \$10 or 16
ESC 1	..... 80 x 24	standard character set
ESC 2	..... 96 x 24	
ESC 3	..... 160 x 24	
ESC 4	..... 80 x 24	high quality character set
ESC 5	..... 80 x 32	

ESC 6	.....	80 x 48
ESC 7	.....	132 x 24
ESC 8	.....	128 x 32

Exit to the Main Menu will return to the default state as set up in the HELLO program for DOS 3.3 or the PARMS file for ProDOS. The same is true of a VID 3 command.

Except for the normal 24 x 80 format, support for the Ultraterm depends on the card being in slot 3.

There may be problems if you try to send things to the printer while in some of the Ultraterm modes. It is recommended that you switch to 40 columns before doing this. Using a PRTR1"<Control-I>80N" command sometimes overcomes the problem.

### MEMORY ALLOCATION WITH MERLIN 8/16

The memory areas \$300-\$3EF in main memory and \$800-\$FFF in auxiliary memory are available for user supplied USER and USR routines. The page 3 area in main memory is intended for I/O interface routines. One cannot send a character to COUT, for example, from auxiliary memory. Merlin does not use these areas. Zero page locations \$90-\$9F are not used by Merlin and are reserved for USER routines (note that the XREF program uses these locations). Zero page locations \$60-\$6F are reserved for user supplied routines and may be used as you wish. No other zero page locations are available.

### CONFIGURING MERLIN 8 (ProDOS)

Configuration data is kept in a file called PARMS which is loaded when the assembler is run. To change the data just change the \+Italic\source\+Italic\ file PARMS.S and reassemble it.

To load the file, set the prefix to /MERLIN 8 and type L to load a source file. Then type SOURCE/PARMS at the prompt. When you are done making changes, reassemble the file. Use S to SAVE the source code as /MERLIN/SOURCE/PARMS. Remember that Merlin 8 adds the .S suffix automatically. Then save the object code as /MERLIN/PARMS by using the O command.

### CONFIGURING MERLIN 8 (DOS 3.3)

The data statements in the Applesoft boot program HELLO contain the configuration information. To change the data just LOAD HELLO, change the data in the DATA statements and SAVE HELLO.

### DATA DESCRIPTION FOR MERLIN 8 CONFIGURATIONS

DATA #	DEFAULT	PURPOSE
1	60	Number of lines per page (for PRTR).
2	0	Lines to skip at page perforation (0 sends a form feed character).
3	80	Number of characters per line (for PRTR).

DATA #	DEFAULT	PURPOSE
4	\$80	Must be \$80 if printer does its own CR at end of line, otherwise should be 0.
5	\$83	80 column flag. Should be \$80+3 if 80 column card is in slot 3 (or Apple 80 col card) is to be selected upon boot. Otherwise 0. MUST BE \$83 WITH ProDOS.
6,7	\$901	Source file start address, must not be less than \$901.
8,9	\$AA00	SHOULD NOT BE CHANGED.
10,11	\$901	End of source pointer. Must equal the Source file start address.
12	\$DE ^^"	The editor's wild card character.
13	4	Number of fields per line in symbol table printout.
14	\$AF "/"	Character searched for by "UPDATE SOURCE" entry to assembler. If this is 0 the question will be bypassed.
15,16,17	14,20,31	The default tabs for editor and assembler, note that these values are relative to the left side of screen.
18	8	Number of object bytes/line after the first line.
19	5	Error/bell flag and Ultraterm start parameters. To disable the bell, set this value to 197. The high bit, if on, will force the assembler to pause forever for a keypress at an error; if off, a sound continues for 20 seconds and then assembly continues. The V bit, if set disables some bells. The low nibble determines the default mode of the Ultraterm if you are using that. The value 5 or \$85 gives the 32X80 mode.
20	\$40	Cursor flag. Gives regular cursor if this is \$40 and block cursor if 0. The Apple 80-col card must have the block cursor and this flag will be overridden if you are using that card.
21	0	LSTDO default: 0,1=LSTDO ON, >1=LSTDO OFF. Bit 0, if clear, causes shift to 40 columns when a PRTR command is issued.
22	72	Column at which the cycle count will be printed when using the CYC opcode.
23	\$EC	Cursor type for Ultraterm. Must be changed if the Ultraterm mode is changed (see byte 19).
24-44	"\$F1 to \$F7"	File type names for the user defined file types \$F1 through \$F7. These names will be shown in the directory when cataloged by Merlin. ProDOS ONLY.

## 64K MERLIN AND MERLIN 8/16 SOURCE FILES

Source files from the original 64k version of Merlin for the Apple II+ can be loaded directly into DOS 3.3 Merlin 8. To use 64k Merlin source files with ProDOS Merlin 8/16 you must use the CONVERT utility supplied with the ProDOS User's Disk. Some changes may be required to the source due to some of the missing pseudo opcodes in Merlin 8/16. If your program uses HIMEM: or SYM, they should be deleted. If your program uses the ERR opcode to check whether SYM or HIMEM: have been set, they should be deleted. If your program uses Sweet 16 then the enabling opcode SW will have to be inserted. Also, any OBJ opcodes will have to be removed since the meaning of this opcode has been changed.

## MERLIN 8/16 ProDOS NOTES

The ProDOS version uses TXT files exclusively for source files. This includes files intended for the PUT or USE opcodes, and all such files must have the .S suffix in the file name, which is automatically appended by Merlin 8/16 for all loads and saves. It is suggested that you keep files intended for PUT or USE in a subdirectory. For example, you could save a file named MYPUT under the pathname LIB/MYPUT. It would then be called in an assembly program by: PUT LIB/MYPUT, or PUT /PREFIX/LIB/MYPUT if the PUT file is in the volume called PREFIX.

If you save a file under a directory name that does not exist, a subdirectory will be created under that name. Suppose you want to save your current source SRC in the volume MYVOL and in the subdirectory SUB which does not exist in the MYVOL directory. Then type /MYVOL/SUB/SRC when the pathname is requested, or just SUB/SRC if /MYVOL/ is the prefix, and the subdirectory SUB will be automatically created and the file SRC placed in it.

It is wise to use a full pathname in operands of the SAV, USES and PUT opcodes, since otherwise the current prefix will be attached to the name and that may not be the prefix you want.

Slot and drive parameters are *\+not\* acceptable by any commands or opcodes. You *\+must\* use pathnames.

Since the ProDOS version of Merlin 8/16 runs under its own interpreter rather than the BASIC interpreter, there is no warm re-entry as with the DOS 3.3 version.

There is no equivalent of the BASIC CAT or CATALOG commands as Disk Commands. The interpreter automatically selects the catalog format for the C command according to whether you are in 40 or 80 column mode on Merlin 8.

The ProDOS volume /RAM/ is disconnected by Merlin 8/16 since it uses all of auxiliary memory.

If Merlin 8/16 cannot find a disk volume required for linking or assembly, it will ask for the correct volume to be inserted. This request can be aborted by pressing Control-C. This only applies to volumes, and not files. Thus, if you want a PUT opcode to prompt you to switch disks, you must use the full pathname with the PUT opcode. Note that this feature will not work with the Linker when using one disk drive.

If the present prefix does not correspond to any volume online, Merlin 8/16 will give a VOLUME NOT FOUND error.

## TRANSFERRING SOURCE FILES FROM DOS 3.3 TO ProDOS MERLIN 8/16

There are two methods of transferring files from the DOS 3.3 versions of Merlin to the ProDOS versions. Since the ProDOS version uses text files only, you could load files into the DOS 3.3 version and write them as text files and then transfer them with Apple's CONVERT program. Unfortunately, CONVERT is not a literal transfer, as it will clear the high bits in the file. The ProDOS version of Merlin 8/16 will set the high bits again, but the tabbing in the editor will be fouled up by this procedure. However, you merely have to type FIX in the editor and resave the source to remedy this problem. Files intended for PUT or USE should be resaved because, otherwise, assembly will be slowed.

Another method is to transfer the files as binary files from DOS 3.3 and use the fact that the ProDOS version of Merlin has the ability to load any type of file including binary files. This does not apply to saving. After loading a binary source file, it should be deleted and saved back as a TXT file. The Load command automatically permits loading of TXT or BIN files. Other types of files can be loaded by changing the byte used to designate source file type which is kept in location \$BE5D which ordinarily holds a 4.

Since the ProDOS version of the assembler does not use the T. prefix of the DOS 3.3 version for PUT files, there will be some renaming of such files that will be necessary.

## MERLIN 8 AND SPEED UP CARDS

Merlin 8 will work either in main or auxiliary memory, aux being the default. If you are using the main memory version, you will get about a 1.6 times speed improvement with the Speedemon card, and about a 2x speed improvement with the Titan Accelerator the Applied Engineering Transwarp card. The difference is due to the speed up of auxiliary memory during assembly.

To select the main memory version of Merlin 8 with DOS 3.3, change the HELLO program to BLOAD MERLIN.X instead of MERLIN.

To select the main memory version of Merlin 8 with ProDOS, use a \$C3 as the fifth byte in the PARMs file. The V-bit of that location is used as a flag to instruct the interpreter to make the main memory modifications.

A plus sign (+) after the Merlin 8 Version number on the Main Menu screen indicates the main memory version is active.

Some utilities do not work with the ProDOS main memory version. This is because ProDOS is moved to auxiliary memory. Programs that do not switch zero pages will work correctly. Programs designed to be run in 64K will most likely run properly. The Filer and Convert programs will run as long as the "-" command is used to run them, and all Merlin 8 utilities will function correctly. The QUIT command moves ProDOS back to main memory.

## MERLIN 8 and DOS 3.3 HARD DISKS or RAM disks

On the DOS 3.3 version of Merlin 8 are files called MERLIN.CORVUS and MERLIN.CORVUS.X. Normally, the DOS 3.3 version of Merlin 8 makes certain patches to DOS 3.3 to allow faster file loading, and the optional cancelling of a Catalog at the screen pause. With hard disks, RAM disks, DOS 3.3 for 3.5" disks, or any other custom version of DOS 3.3, the standard version of Merlin 8 may not work properly. The files MERLIN.CORVUS and MERLIN.CORVUS.X are versions of Merlin 8 which do not modify DOS 3.3 in any way, and are thus compatible with the Corvus and other custom DOS 3.3 software. To use the file called MERLIN.CORVUS, rename the file called MERLIN to TEMP.MERLIN and then rename MERLIN.CORVUS to MERLIN. MERLIN.CORVUS.X is a main memory version of MERLIN.CORVUS, which you may want to use if you are using the SpeedDemon accelerator card instead of MERLIN.X.

## CONFIGURING MERLIN 16

Merlin 16 can be customized by re-assembling the file PARMS.S in the SOURCE directory of the Merlin 16 diskette. The object file, PARMS, must be saved to the same directory that Merlin.System is located in.

## THE MERLIN 16 PARMS FILE

Here is a listing of the PARMS.S file for Merlin 16. By examining the comments, you can see which attributes of the assembler can be changed.

```

1 *=====
2 *  PARMS for Merlin.16
3 *-----
4
5         TR      ADR
6         TR
7
8 Y       EQU     1
9 y       EQU     1
10 N      EQU     0
11 n      EQU     0
12
13 SAVOBJ  KBD     "Save object code? (Y,N) "
14
15         ORG     $8000
16
17 DATA   ORG     $E4F3
18
19         DFB     60          ;# lines/page for PRTR
20         DFB     0           ;Page skip (formfeed if 0)
21         DFB     80          ;# printer columns
22         DFB     $80         ;- if printer does CR at

```



```

23                                     ; end of # columns
24      ORG
25      ORG    $E009
26
27      DFB    $83      ;80 col flag (DO NOT CHANGE
28                                     ; except V-bit which will
29                                     ; cause ProDOS to be moved
30                                     ; to aux memory and Merlin
31                                     ; to load into main memory.
32                                     ; I.e., use $C3 for this.)
33
34 *-----
35 * Source address must be $901 or above.
36 * It can be set higher to protect an area
37 * of RAM above $900 for any purpose:
38 *-----
39
40 SOURCE    =    $901
41
42      DA    SOURCE    ;Start of source
43      DA    $9E00     ;Reserved
44      DA    SOURCE    ;End of source
45
46      DFB    0        ;main menu accepts RTN as "Y"
47                                     ; if this is set to $FF
48      DFB    4        ;# of symbol columns
49
50 *-----
51 * Following flags byte has all bits significant:
52 *
53 * Bit 7 = print date on page header of assembly.
54 * Bit 6 = return to main menu after a key press
55 *   at end of an assembly, or to full screen editor
56 *   at point of error if an assembly abort occurs
57 *   (or also if REL is active). If this bit is 0,
58 *   and bit 0 is 1, then it goes to the command line
59 *   editor.
60 * Bit 5 = linkers should check auxtype bit 0 of
61 *   source files to decide whether to do assemblies.
62 * Bits 4,3 = default XC mode:
63 *   00 = defaults to 6502
64 *   10 = 65C02 mode
65 *   11 = 65816 mode
66 *   (Do not use other values.)
67 * Bits 2,1 = default MX mode on entry:
68 *   00 = full 16-bit mode on entry
69 *   10 = short M, long X
70 *   01 = long M, short X
71 *   11 = 8 bit mode
72 * Bit 0, if set, enables command line editor access.
73 *-----
74
75      DFB    %11111110 ;Misc flags, see above

```

```

76
77         DFB  9,15,26      ;Default tabs
78
79         DFB  4            ;# obj bytes/line after 1st
80
81 *-----
82 * Following flags byte has 3 high bits significant:
83 *
84 * Bit 7 = wait forever for key upon assembly error.
85 * Bit 6 = defeat most bells if set.
86 * Bit 5 = do not pause or sound alarm on an assembly
87 *         error (must also have bit 7 clear for this).
88 *
89 * Low nibble is Ultraterm entry mode:
90 *
91 * Eg., $05, $45, $85, etc give 32x80 interlace mode.
92 *-----
93
94         DFB  $05          ;Bell flags & UT mode.
95
96         DFB  %01000000    ;Upper case convert mode
97                             ; when entering full screen
98                             ; editor if negative.
99                             ; (Conversion is done for
100                            ; label, opcode and operand
101                            ; fields only and only when
102                            ; tabs are not zeroed.)
103                            ;V-bit = default cursor
104                            ; mode, 1=insert cursor
105
106 *-----
107 * Following flags byte has 5 significant bits:
108 *
109 * Bit 7 = assembler is label case insensitive if on.
110 * Bit 6 = defeat screen ed screen blank, if on.
111 * Bit 5 = enable list of local labels, if on.
112 * Bit 1 = LSTDO default, DO off areas not listed if
113 *         this bit is on (and default not overridden).
114 * Bit 0 = defeat shift to 40 columns on PRTR1, if
115 *         bit is on (shift only occurs when Ultraterm
116 *         is active with more than 24 rows, so it is
117 *         best to leave this bit as is (off)).
118 *-----
119
120         DFB  %01000000    ; case sensst., no blanking
121
122         DFB  80-8         ;Column for cycle count
123
124         ORG  $B23E
125
126
127         DFB  %9F&"["      ;Catalog abort key, now ESC
128

```

```

129         ORG
130
131         ERR    *-DATA-23 ;23 data bytes to here.
132
133 * User file type names:
134
135         ORG    $B6B6      ;Adrs subject to change
136
137         ASC    "$F1"
138         ASC    "$F2"
139         ASC    "$F3"
140         ASC    "$F4"
141         ASC    "$F5"
142         ASC    "$F6"
143         ASC    "$F7"
144
145         ORG
146
147         ERR    *-DATA-44
148
149 *-----
150 * Screen editor variable parameters:
151 * Cursors are: insert, find in insert mode,
152 * overstrike, find in overstrike mode.
153 *-----
154
155         ORG    $DFBC
156
157 CURSORS  INV    'IF F'
158
159         DFB    $A0        ;Cursor blink rate
160
161 *-----
162 * Screen editor cmd chars:
163 * The cursor keys (although here) must NOT
164 * be changed or the editor will not work
165 * correctly.
166 *-----
167
168         DFB    $9F&"U"    ;Don't change
169         DFB    $88        ; "
170         DFB    $9F&"I"    ;Tab key insert toggle
171         DFB    $9F&"T"    ;^T (remember this place)
172         DFB    $9F&"B"    ;Go to line beginning
173         DFB    $9F&"N"    ;Go to line end
174         DFB    $9F&"R"    ;Cancel changes
175         DFB    $9F&"S"    ;Status box
176         DFB    $9F&"F"    ;Find char
177         DFB    $9F&"W"    ;Next word
178         DFB    $9F&"L"    ;Toggle uc/lc auto shift
179         DFB    $9F&"D"    ;Delete char under cursor
180         DFB    $FF        ;Delete previous char
181         DFB    $9F&"Y"    ;Clear to end-of-line

```

```

182         DFB  $9F&"O"    ;Accept next key literal
183
184 * Open-apple key cmds, must be upper case:
185
186         DFB  #"X"        ;Cut
187         DFB  #"C"        ;Copy
188         DFB  #"V"        ;Paste
189         DFB  #"F"        ;Find
190         DFB  #"W"        ;Find word
191         DFB  #"E"        ;Exchange text
192         DFB  #"T"        ;Go to ^T selected point
193         DFB  $9F&"I"    ;Insert line at cursor
194         DFB  #"I"        ; "
195         DFB  $9F&"["    ;Go to cmd line ed if enabled
196         DFB  #"B"        ;Go to beginning
197         DFB  #"N"        ;Go to end
198         DFB  $DF        ;Delete preceding line (DEL)
199         DFB  #"L"        ;Locate text
200         DFB  #"Z"        ;Center line with cursor
201         DFB  $9F&"J"    ;Don't change
202         DFB  $9F&"K"    ; "
203         DFB  #"Y"        ;Select from here on
204         DFB  $9F&"U"    ;Don't change
205         DFB  $9F&"H"    ; "
206         DFB  #"D"        ;Delete this line
207         DFB  #"8"        ;Line of asterisks
208         DFB  #"9"        ;Line bordered by asterisks
209         DFB  #"-"        ;Line of dashes
210         DFB  #"="        ;Line of equal signs
211         DFB  #"1"        ;PRTR1 + assemble
212         DFB  #"2"        ;PRTR1 + USER + assemble
213         DFB  #"3"        ;PRTR3 + assemble
214         DFB  #"4"        ;PRTR3 + USER + assemble
215         DFB  #"A"        ;ASM command
216         DFB  #"Q"        ;Quit to main menu
217         DFB  #"H"        ;Toggle split screen
218         DFB  #"O"        ;Open command box
219         DFB  #"6"        ;LINK (for LINKER.GS only)
220         DFB  #"R"        ;Replace line
221
222         ORG
223
224         ERR  *-DATA-99
225
226 *=====
227 * Printer init string, used when PRTR
228 * issued with empty first string.
229 * A CR is always issued after the
230 * string, so none need be here.
231 *-----
232
233         ORG  $DBF0
234

```

```
235          DS    15          ;PRTR init default string
236
237          ORG
238
239          ERR    *-DATA-114
240
241 *=====
242 *   Default STARTUP file:
243 *-----
244
245          ORG    $2006
246
247 STUP     STR    "LINKER.GS" ;You can change this string
248
249          DS    $40-.*+STUP ; but not this line
250
251          ORG
252
253          ERR    *-DATA-178
254
255 *=====
256 * The PARMS file must be in the
257 * SAME DIRECTORY as MERLIN.SYSTEM
258 *-----
259
260          DO     SAVOBJ
261          SAV   PARMS
262          FIN
```

## Screen Blanking

The PARMS file for Merlin 16 supports an optional setting for telling Merlin 16 to blank the screen if nothing is typed for an extended period. Pressing any key, such as Return, will restore the normal screen display.

This is intended to protect your monitor screen in situations where the computer is left unattended for long periods of time, and Merlin 16 is the only program generally run on that computer. Any video image will tend to "burn" itself into a monitor screen if left on continuously for several hours a day, every day, for a period of many months. Note this is not a specific problem to Merlin 8/16.

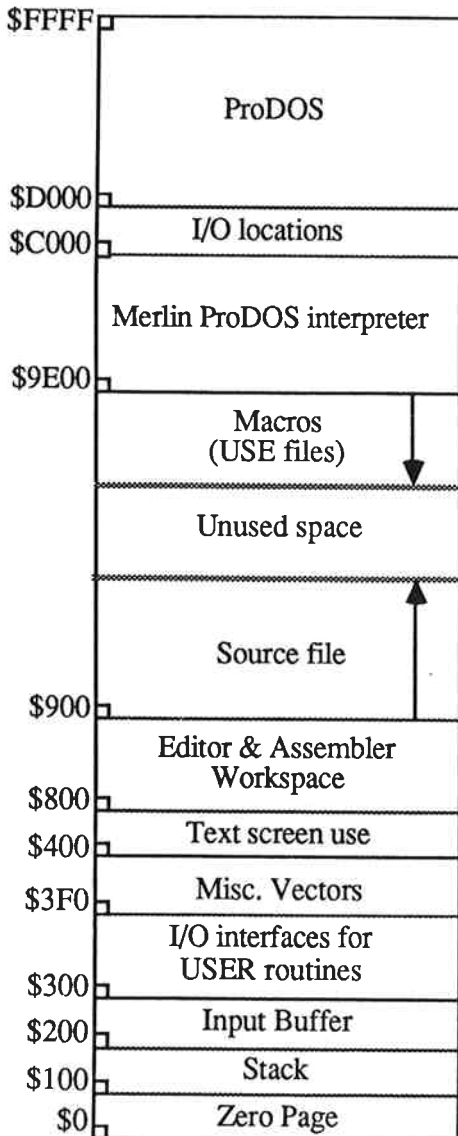
You should not concern yourself with this unless you are in a working environment where the Merlin 16 menu is left on all day when the computer is unattended.

Because it can be disconcerting to have the screen suddenly go blank if you are not aware of this feature, Merlin 16 is shipped with this feature disabled. To enable screen blanking, set the high bit of the flag near line 120 of the Merlin 16 PARMS.S file. See the comments in the source listing for details.

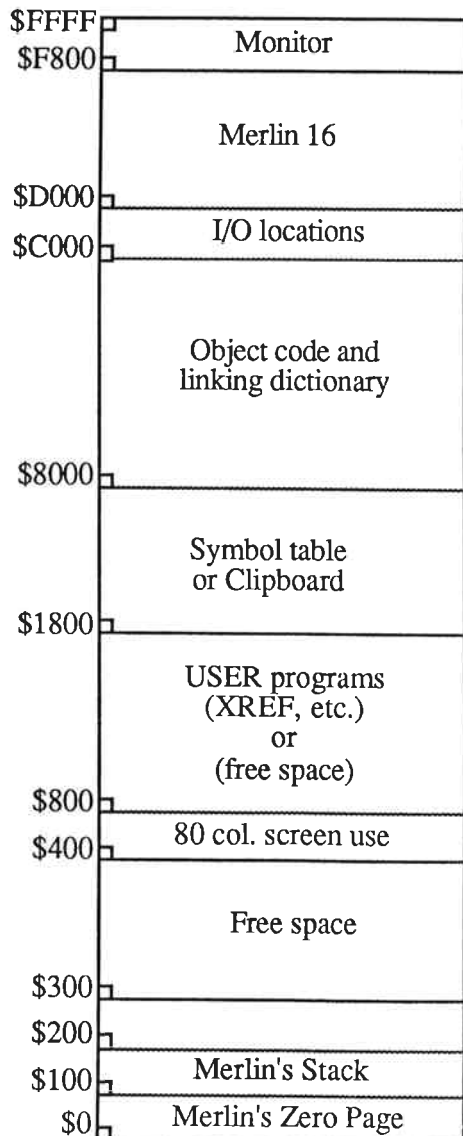
## MERLIN 8/16 MEMORY MAPS

## Merlin 16 - ProDOS

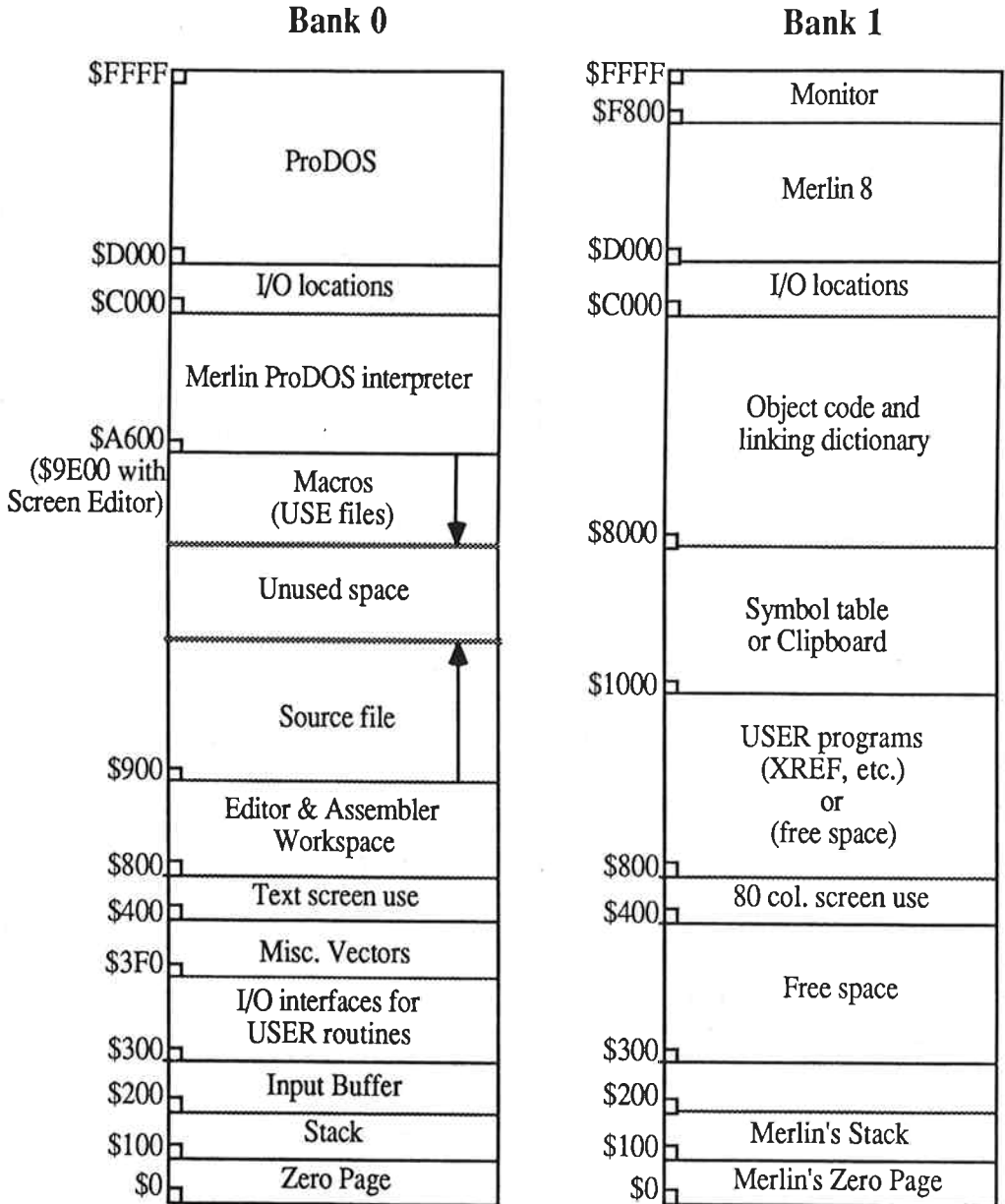
## Bank 0



## Bank 1

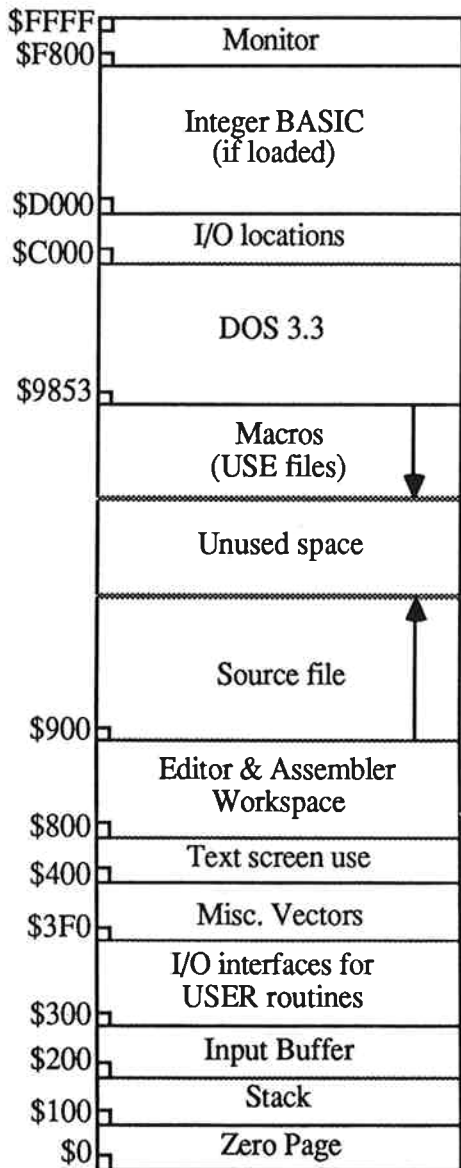


## Merlin 8 - ProDOS

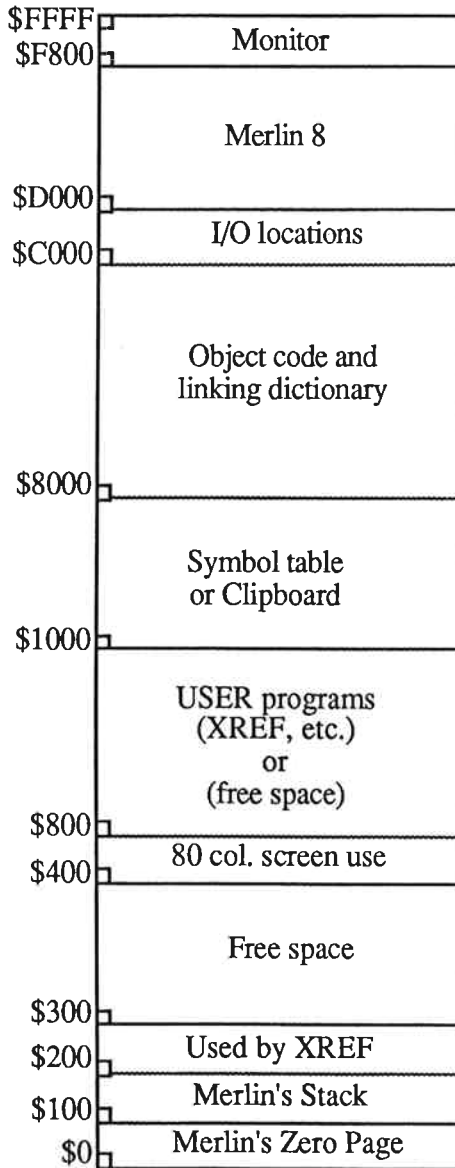


### Merlin 8 - DOS 3.3

#### Bank 0



#### Bank 1





---

## ERROR MESSAGES

### BAD ADDRESS MODE

The addressing mode is not a valid 6502 instruction; for example, JSR (LABEL) or LDX (LABEL),Y.

### BAD BRANCH

A branch (BEQ, BCC, etc.) to an address that is out of range, i.e. further away than +127 bytes.

**NOTE:** Most errors will throw off the assembler's address calculations. Bad branch errors should be ignored until previous errors have been resolved.

### BAD EXTERNAL

EXT or ENT in a macro or an equate of a label to an expression containing an external, or a branch to an external (use JMP).

### BAD INPUT

This results from either no input, i.e. Return alone or an input exceeding 37 characters in answer to the KBD opcode's request for the value of a label.

### BAD LABEL

This is caused by an unlabeled EQU, MAC, ENT or EXT, or a label that is greater than 13 characters, or one containing illegal characters. A label must begin with a character at least as large in ASCII value as the colon and may not contain any characters less than the digit zero.

### BAD OBJ

An OBJ after code start or OBJ not within \$4000 to \$BFEO.

### BAD OPCODE

Occurs when the opcode is not valid, or misspelled, or the opcode is in the label column.

### BAD ORG

Results from an ORG at the start of a REL file.

**BAD PUT**

This is caused by a PUT inside a macro or by a PUT inside another PUT file.

**BAD REL**

A REL opcode occurs after some labels have been defined.

**BAD SAV**

This is caused by a SAV inside a macro or a SAV after a multiple OBJ after the last SAV.

**BAD VARIABLE**

This occurs when you do not pass the number of variables to a macro that the macro expects. It can also occur for a syntax error in a string passed to a macro variable, such as a literal without the final quote.

**BREAK**

This message is caused by the ERR opcode when the expression in the operand is found to be non-zero.

**DICTIONARY FULL**

Overflow of the relocation dictionary in a REL file.

**DUPLICATE SYMBOL**

On the first pass, the assembler finds two identical labels.

**FILE TYPE MISMATCH**

A file specified to be loaded, such as a source file, or REL file during a Link, does not have the expected file type.

**FORMAT ERROR**

A command has been incompletely typed, for example, typing PRTR instead of PRTR 1.

**ILLEGAL CHAR IN OPERAND**

A non-math character occurs in the operand where the assembler is expecting a math operator. This usually occurs in macro calls with improper syntax resulting from the textual substitution.

**ILLEGAL FILE TYPE (ProDOS version only)**

TYP opcode used with an illegal operand.

**ILLEGAL FORWARD REFERENCE**

A label equated to a zero page address after it has been used. This also occurs on the first pass when an unknown label is used for some things that must be able to calculate the value on the first pass, e.g. ORG< OBJ DUM. It also occurs if a label is used before it is defined in a DUM section on zero page.

**ILLEGAL RELATIVE ADRS**

In REL mode a multiplication, division or logical operation occurs in a relative expression. In Merlin 8, this also occurs for an operand of the type #>expr or a DFB >expr when the expr contains an external and the offset of the value of the expr from that of the external exceeds 7.

**MEMORY FULL Errors**

There are three common causes for the MEMORY FULL error message.

**MEMORY FULL IN LINE: xx.** Generated during assembly.

**CAUSE #1:** Too many symbols in the symbol table, causing it to exceed available space.

**REMEDY #1:** Make the symbol table larger by setting OBJ to \$BFE0 and use DSK to assemble directly to disk.

**CAUSE #2:** If the combined size of the source file and a PUT file is too large.

**REMEDY #2:** Split either file into two smaller files.

**ERR:MEMORY FULL.** Generated immediately after you type in one line too many.

**CAUSE:** The source code is too large and has exceeded available RAM.

**REMEDY:** Break the source file up into smaller sections and bring them in when necessary by using the PUT pseudo-op.

**ERROR MESSAGE:** None, but no object code will be generated. There is no OBJECT information displayed on the Main Menu.

**CAUSE:** Object code generated from an assembly would have exceeded the available 16K space.

**REMEDY:** Set OBJ to an address less than its \$8000 default or use the DSK pseudo-op.