



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 `\+Italic\not\~Italic\` acceptable by any commands or opcodes. You `\+Italic\must\~Italic\` 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 PARM.S in the SOURCE directory of the Merlin 16 diskette. The object file, PARM.S, must be saved to the same directory that Merlin.System is located in.

THE MERLIN 16 PARM.S FILE

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

```

1 *=====
2 *  PARM.S 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
125         ORG    $B23E
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.

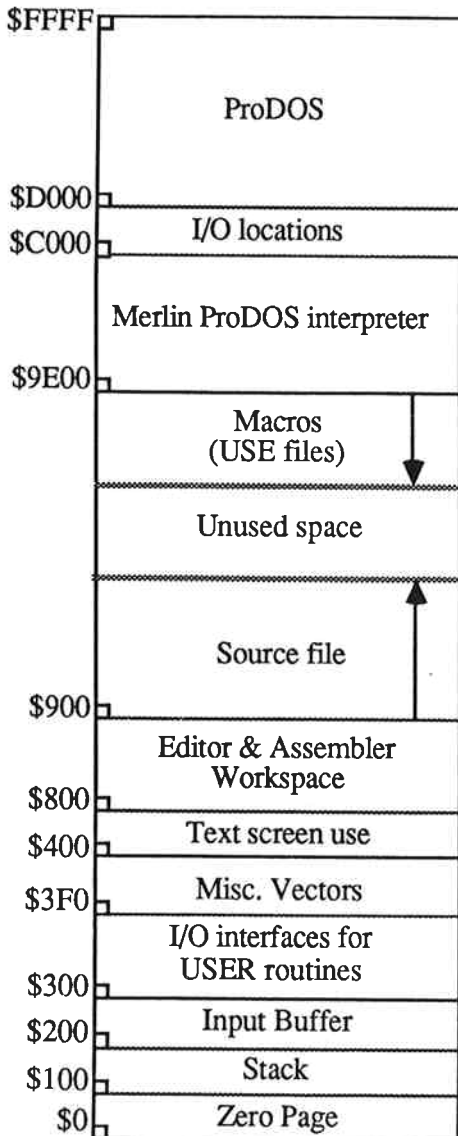
You should not concern yourself with this unless you are in a working environment where the Merlin 16 menu is left on all day when the computer is unattended.

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.

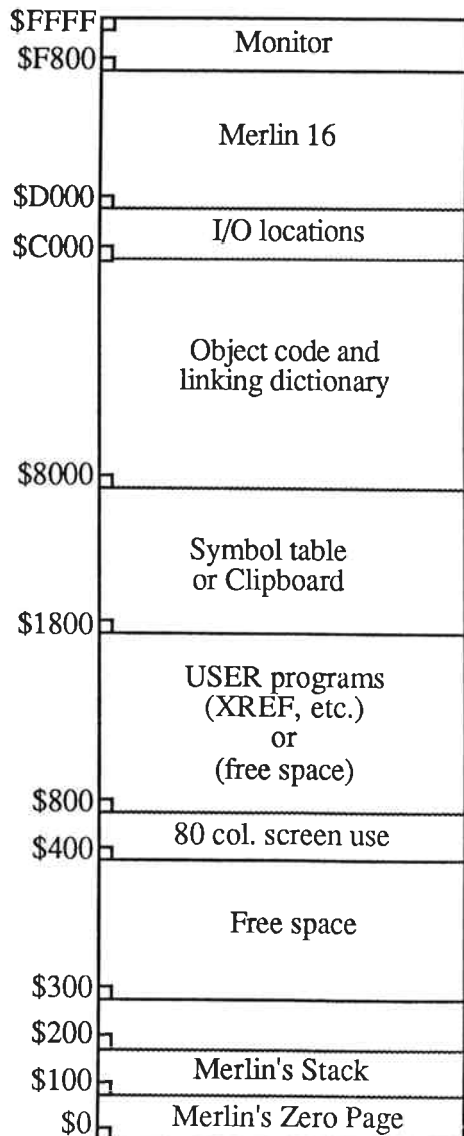
MERLIN 8/16 MEMORY MAPS

Merlin 16 - ProDOS

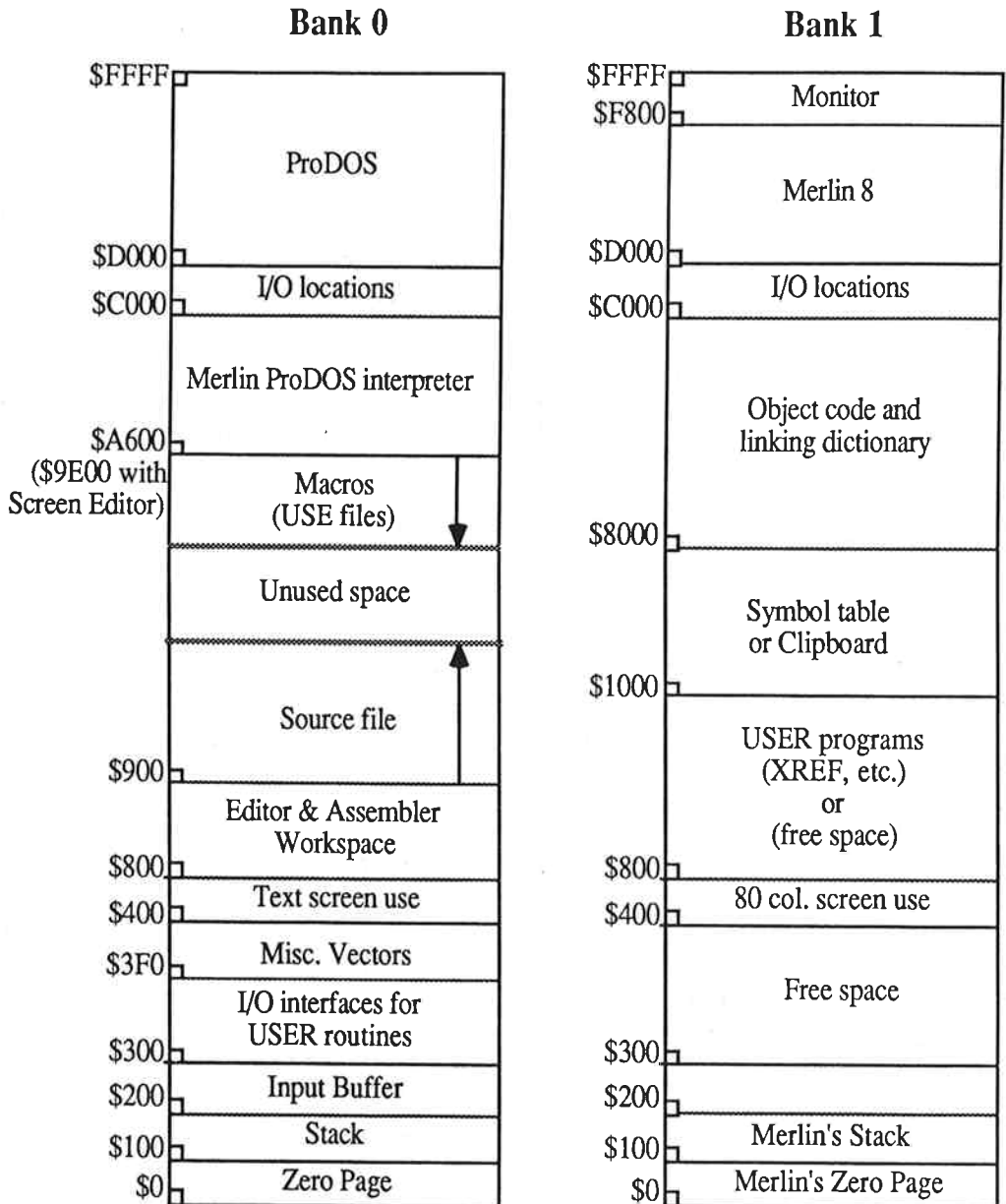
Bank 0



Bank 1

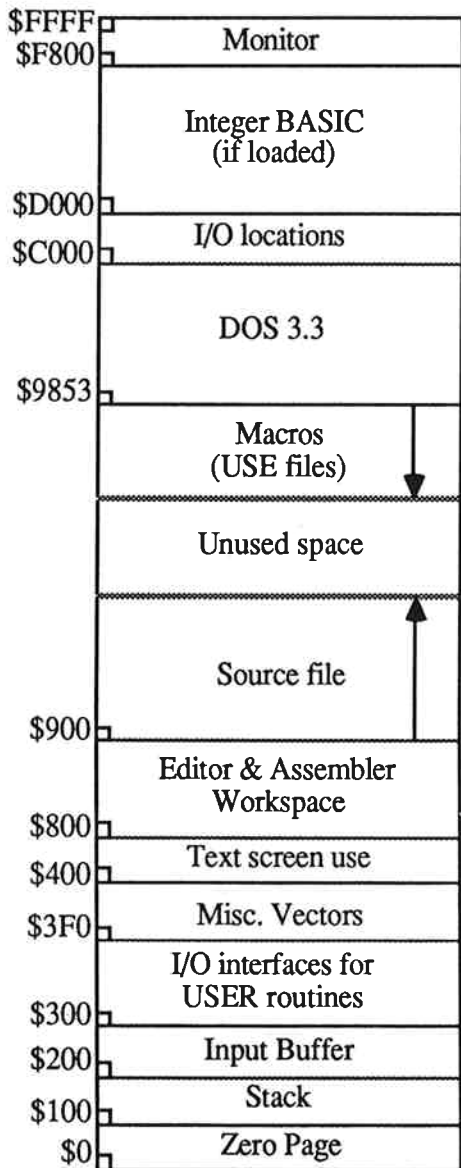


Merlin 8 - ProDOS

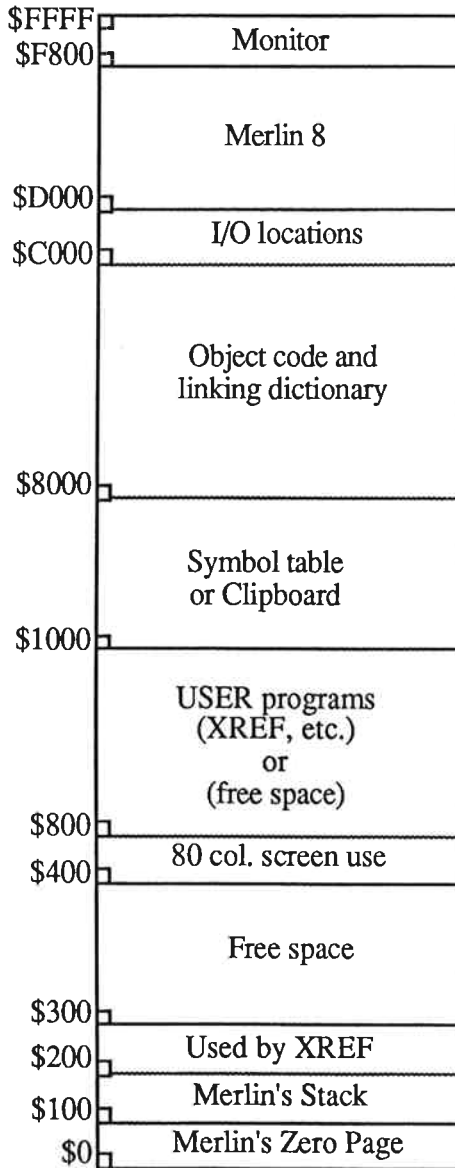


Merlin 8 - DOS 3.3

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.

MEMORY IN USE

This error results from trying to load a file into a part of memory already in use. This could be by loading a tool-large PUT file, appending a file without sufficient remaining memory, or loading a USE library. It can also occur from loading a USER function when one is already in memory. In Merlin 8, the error occurs if you try to load a utility like SOURCEROR when the Editor (ED) is already loaded. Use REMOVE.ED before loading another utility.

MISALIGNMENT - MERLIN 16

This means that the value of a label on the second pass differs from what it was on the first pass and should indicate a forward reference that the assembler could not resolve. The most common cause is a forward reference to a zero page equate or dummy section. Misalignment may also be caused by a previous error. A misalignment error is given only once in order to avoid the error routine on all subsequent labels, all of which are likely to be misaligned.

NESTING ERROR

Macros nested more than 15 deep or conditionals nested more than 8 deep will generate this error.

NOT A MACRO

A Macro name has been used that has not been previously defined.

NOT RESOLVED

This is a Linker error that alerts you to the fact that an EXT label reference in one of the linked files could not be found in any of the other linked files.

OBJECT SPACE OVERFLOW - MERLIN 16 ONLY

If the available object file space has been exceeded, the assembler prints this message. This disables the Object file save command at the Main Menu.

OUT OF MEMORY

This is a memory error from the Full Screen Editor. It is caused by trying to cut or copy more text to the clipboard, or to paste more text into the document than there is free memory to accommodate.

RANGE ERROR

When saving a source file larger than 32K, you may get this error, in particular if are using a standard DOS 3.3 and have not booted on the Merlin 8 DOS 3.3 diskette. This is due to a limitation of DOS itself. The problem can be temporarily fixed however by changing byte \$A964 in memory from \$7F to \$FF. (For those experienced with a disk sector editor, you can change Track 1, Sector 8, Byte \$64 from \$7F to \$FF to make this change permanent on whatever DOS 3.3 disk you wish).

SYNTAX ERROR

The proper syntax has not been used in a command, either because of typing errors, or extra or omitted characters or parameters.

TWO EXTERNALS

Two or more externals in an operand expression.

UNKNOWN LABEL

Your program refers to a label that has not been defined. This also occurs if you try to reference a MACRO definition by anything other than PMC or >>>. It can also occur if the referenced label is in an area with conditional assembly OFF. The latter will not happen with a MACRO definition.

256 EXTERNALS

The file has more than 255 externals.

NOTE: When an error occurs that aborts assembly, the line containing the error is printed to the screen. This may not have the same form as it has in the source, since it shows any textual substitutions that may have occurred because of macro expansion. If it is in a macro call, the line number will be that of the call line and not of the line in the macro which is unknown to the assembler.

SOURCEROR

Sourceror is a sophisticated and easy to use co-resident disassembler designed to create Merlin 8/16 source files out of binary programs, usually in a matter of minutes. There are two versions of Sourceror in Merlin 8/16. Sourceror on the Merlin 16 disk disassembles 6502, 65C02, 65802 and 65816 object code. Sourceror on the Merlin 8 diskettes disassembles 6502, 65C02, and 65802 object code and is not designed for use on the IIGs.

USING SOURCEROR ON MERLIN 16

To use Sourceror on the Merlin 16 diskette, follow these steps:

1. From the Main Menu, press D for Disk Command.
2. At the Disk Command prompt type: BRUN SOURCEROR/OBJ
3. Press F to enter the Editor.
4. Press Open-Apple-O to open the Command Box.
5. Type DIS "MYFILE" where MYFILE is the name of the object file to be disassembled.

The DIS command is available in Merlin 16 only and uses the following syntaxes:

DIS "MYFILE" [disassemble MYFILE at the original running address]

DIS \$1000"MYFILE" [disassemble MYFILE at the specified address of \$1000]

SYS files are always assumed to have a running address of \$2000. If a non-zero address is specified in the DIS command, that address is taken as the initial running address. If the address is not specified, the Merlin 16 Sourceror uses the information in the AUXTYPE field for the running address.

USING SOURCEROR ON MERLIN 8

NOTE: On the ProDOS version of Merlin 8, the Full Screen Editor uses the same part of memory as Sourceror. Therefore you will have to temporarily deactivate the Full Screen Editor. This is not necessary with the DOS 3.3 version of Merlin 8.

To deactivate the Full Screen Editor on Merlin 8 ProDOS version:

1. From the Main Menu, press D for Disk Command.

2. At the Disk Command prompt type: BRUN /MERLIN.8/UTILITIES/REMOVE.ED

To use Sourceror on the Merlin 8 diskette, follow these steps:

1. Merlin 8 DOS 3.3 version

From the Main Menu, press C to Catalog.

At the Command prompt, type BRUN SOURCEROR and press Return.

Merlin 8 ProDOS version

After deactivating the editor as shown above, you should be at the Disk Command prompt.

At the Disk Command prompt, type BRUN /MERLIN.8/SOURCEROR/OBJ and press Return.

2. Merlin 8 DOS 3.3 version

Press E to enter the Editor.

Merlin 8 ProDOS version

Press Return to go back to the Main Menu.

Press E to enter the Editor.

3. Press Escape Control-Q to set the screen to 40 columns.

The USER command will be ignored and Sourceror will not run if the 80 column screen is in effect. Do not use Escape 4 to change to 40 columns.

4. From the 40 column screen, type USER and press Return.

5. You will be prompted to "Load an object file?"

If you have loaded the object file prior to using Sourceror, press N.

If you have not loaded an object file yet, press Y and enter the filename. It will be loaded showing the load address and end of program address. After you have noted these addresses, press any key.

NOTE: If you type Control-S after the filename to be loaded, files using a RAM version of SWEET 16 can be disassembled.

7. If you have loaded a file prior to using Sourceror you will be prompted regarding the location of the object code.

Press Return if the program to be disassembled is at its original (running) location.

If not, you must specify in hex the present location of the code to be disassembled. You will also be prompted to give the original location of that program.

8. The Sourceror Main Menu appears.

Your first command must include a hex address. Thereafter this is optional, as explained shortly. You may now start disassembling or use any of the other commands.

NOTE: When disassembling, you must use the *original* address of the program, not the address where the program currently resides. It will appear that you are disassembling the program at its original location, but Sourceror is actually disassembling the code at its present location and translating the addresses.

65C02 OPCODES ON OLDER IIE/IIC COMPUTERS

To disassemble 65C02 codes with the older Iie/Iic ROMs, you must first BRUN the file called MON.65C02 on the Merlin 8 diskette.

If you are using Merlin 8 DOS 3.3, before using Sourceror, you must quit to BASIC from the Main Menu, then type BRUN MON.65C02 and press Return.

For the ProDOS version of Merlin 8, type in D for Disk Command at the Main Menu, and then BRUN /MERLIN/UTIL/MON.65C02 and press Return.

See the information on using the XC opcode for details on assembling 65C02 programs.

This utility is not needed with the enhanced Iie or Iic (Unidisk 3.5 compatible) ROMs.

DISASSEMBLY COMMANDS

Your first command must include a hex address. Therefore, if you wanted to list the first 20 lines and the starting address was \$8000, at the \$ prompt you would type 8000L and press Return. All commands accept a 4-digit hex address before the command letter. If this number is omitted after the first command, the disassembly continues from its present address.

If you specify a number greater than the present address, a new ORG will be created.

More commonly, you will specify an address less than the present default value. In this case, the disassembler checks to see if this address equals the address of one of the previous lines. If so, it simply backs up to that point. If not, then it backs up to the next used address and creates a new ORG. Subsequent source lines are erased. It is generally best to avoid new ORGs when possible. If you get a new ORG and don't want it, try backing up a bit more until you no longer get a new ORG upon disassembly.

This backup feature allows you to repeat a disassembly if you have, for example, used a HEX or other command, and then change your mind.

H (Hex)

This creates the HEX data opcode. It defaults to one byte of data. If you insert a one byte hex number using one or two digits after the H, that number of data bytes will be generated.

L (List)

This is the main disassembly command. It disassembles 20 lines of code. It may be used in multiples, thus 2000LLL will disassemble 60 lines of code starting at \$2000. In the Merlin 8 Sourceror, if a JSR to the SWEET 16 interpreter is found, disassembly is automatically switched to the SWEET 16 mode.

The L command always continues the present mode of disassembly, SWEET 16 or normal.

NOTE: If an illegal opcode is encountered, the bell will sound and opcode will be printed as three question marks in flashing format. This is only to call your attention to the situation. In the source code itself, unrecognized opcodes are converted to HEX data, but are not displayed on the screen.

N (Normal - Merlin 8 only)

This is the same as L, but forces disassembly to start in normal 6502 mode.

O (Org - Merlin 16 only)

This command can be used to change the ORG on the fly during disassembly. It is useful for programs that move code to other locations after loading. The syntax for this command is address O newaddress. Do not use any spaces in this command syntax, and note that it uses the letter O and not zero.

ADRSONEWADRS [backup to ADRS and place and ORG to NEWADRS there]

Therefore a command of 1045O300 followed by a Return would tell Sourceror to backup to address 1045 and place an ORG \$300 there.

Q (Quit)

This ends disassembly and goes to the final processing which is automatic. If you type an address before the Q, the address pointer is backed to, but does not include, that point before the processing. For example, if at the end of the disassembly, the lines included:

```
2341- 4C 03 E0      JMP $E003
2344- A9 BE 94      LDA $94BE,Y
```

and the last line was garbage, you could type 2344Q and press Return. This would cancel the last line, but retain all the previous lines.

R (Read)

In Merlin 8, this allows you to look at memory in a format that makes imbedded text stand out. For example, to look at the data from \$1000 to \$10FF, you would type 1000R and press Return. After that, R and Return will bring up the next page of memory. The numbers you use for this command are totally independent of the disassembly address.

You may also disassemble, then use (address)R and Return, then L and Return, and the disassembly will proceed just as if you never used R at all. If you don't intend to use the default address when you return to disassembly, it may be wise to note where you wanted to resume, or to use the / command before the R command. Merlin 16 prints ASCII during disassembly.

S (SWEET 16 - Merlin 8 only)

This is similar to L, but forces the disassembly to start in SWEET 16 mode. SWEET 16 mode returns to normal 6502 mode whenever the SWEET 16 RTN opcode is found.

T or TT (Text)

This attempts to disassemble the data at the current address as an ASCII string. Depending on the form of the data, this will automatically be disassembled under the pseudo-opcode ASC, DCI, INV or FLS. The appropriate delimiter (" or ') is automatically chosen. The disassembly will end when the data encountered is inappropriate, or when 62 characters have been treated, or when the high bit of the data changes. In the last condition, the ASC opcode is automatically changed to DCI.

Sometimes the change to DCI is inappropriate. This change can be defeated by using TT instead of T in the command.

Occasionally, the disassembled string may not stop at the appropriate place because the following code looks like ASCII data to Sourceror. In this event, you may limit the number of characters put into the string by inserting a one or two digit hex number after the T command.

T or TT may also have to be used to establish the correct boundary between a regular ASCII string and a flashing one. It is usually obvious where this should be done.

W, WW, or W- (Word)

This disassembles the next two bytes at the current location as a DA opcode. Optionally, if the command WW is used, these bytes are disassembled as a DDB opcode.

If W- is used as the command, the two bytes are disassembled in the form DA LABEL-1. The latter is often the appropriate form when the program uses the address by pushing it on the stack. You may detect this while disassembling, or after the program has been disassembled. In the latter case, it may be to your advantage to do the disassembly again with some notes in hand.

/ (Cancel)

This essentially cancels the last command. More exactly, it re-establishes the last default address (the address used for a command not necessarily attached to an address). This is a useful convenience which allows you to ignore the typing of an address when a backup is desired.

As an example, suppose you type T to disassemble some text. You may not know what to expect following the text, so you can just type L to look at it. Then if the text turns out to be followed by some HEX data such as \$8D for a carriage return, simply type / to cancel the L and type the appropriate H command.

MERLIN 16 SOURCEROR NOTES

Sourceror on the Merlin 16 disk is fully 65816 compatible. It attempts to follow the flow of the program in assigning the length of immediate operands, thus changing REP, SEP, and XCE when appropriate. Since this cannot always be successful, you can reset this mode by using an address with the L command. When an address is used with the L command, the M and X status bits are taken from the byte following the L. The default is MX = 00. Thus, 1012L01 and Return would restart the disassembly at \$1012 with a long M and a short X, and MX would be set to 01.

The reversed DDB is not supported in the Merlin 16 version of Sourceror. Instead, you can use the WW command to generate a 4-byte long address using the psuedo-op ADRL.

The Merlin 16 Sourceror only uses the 80 column screen.

SOURCEROR XL

For large object files, there is an alternative version of Sourceror on the Merlin 16 disk. To load it from the Main Menu Disk Command prompt, you would type BRUN SOURCEROR/XL and press Return. This version disassembles to disk instead of memory. The files are placed in the current prefix directory active at the time the DIS command is issued.

FINAL PROCESSING

After the Q command, the program does some last minute processing of the assembled code. If you press Reset at this time, you will return to Merlin 8/16 and lose the disassembled code.

The processing may take from a second or two for a short program and up to several minutes for a long one. Be patient.

When the processing is done, you are returned to Merlin 8/16 with the newly created source in the Editor. You can use the Editor to edit or assemble the listing. After a successful assembly, you can save the new code with the Main Menu Save Source and Save Object Code commands.

NOTE: If you are using the ProDOS version of Merlin 8, you can enter the Editor and type USER1 to get rid of Sourceror and free up the memory used by the disassembler. This is not necessary in the DOS 3.3 version.

MODIFYING THE FINISHED SOURCE

In most cases, after you have some experience and assuming you used reasonable care, the source will have few, if any, defects.

You may notice that some DA's would be more appropriate in a DA LABEL-1 or a DDB LABEL format. In such cases, it may be best to do the disassembly again with some notes in hand. The disassembly is so quick and painless that it is often much easier than trying to alter the source directly.

The source will have all the exterior or otherwise unrecognized labels at the end in a table of equates. You should look at this table closely. It should not contain any zero page equates except ones resulting from DA's, JMP's or JSR's. This is almost a sure sign of an error in the disassembly (yours, not Sourceror's). It may have resulted from an attempt to disassemble a data area as regular code.

NOTE: If you try to assemble the source under these conditions, you will get an error as soon as the equates appear. If, as eventually you should, you move the equates to the start of the program, you will not get an error, but the assembly *may not be correct*.

It is important to deal with this situation immediately since trouble could occur if, for example, the disassembler finds the data AD008D. It will disassemble it correctly, as LDA \$008D. The assembler always assembles this code as a zero page instruction, giving the two bytes A5 8D. Occasionally you will find a program that uses this form for a zero page instruction. In that case, you will have to insert a character after the LDA opcode to have it assemble identically to its original form. Since it was data in the first place rather than code, it must be dealt with to get a correct assembly.

THE MEMORY FULL MESSAGE

When the source file reaches within \$600 bytes of the end of its available space you will see MEMORY FULL and be prompted to HIT A KEY. Sourceror will then go directly to the final processing. The reason for the \$600 byte gap is that Sourceror needs a certain amount of space for this processing. There is an optional override provision at the memory full point. If you press Control-O for override, then Sourceror will return for another command. You can use this to specify the desired ending point. You can also use it to go a little further than Sourceror wants you to, and disassemble a few more lines. Obviously, you should not carry this to extremes. If you get too close to the end of available space, Sourceror will no longer accept this override and will automatically start the final processing.

CHANGING SOURCEROR'S LABEL TABLES

The label tables used by Sourceror are just assembled Merlin 8/16 source files. The source file is called LABEL.S and is on the Merlin 8/16 diskettes. It can be modified directly by the user. It must be assembled and saved under the same name as the previous label file, i.e. you have to replace the old existing file.

If you have several label tables you wish to use, you may want to just rename them. For example, you could keep TABLE.DOS, TABLE.PRODOS, etc. on the disk, and then just rename the file as you needed it to LABELS.

APPLESOFT SOURCE LISTING

SOURCEROR.FP

A fully labeled and commented source listing of Applesoft BASIC can be generated by the program called Sourceror.FP which is on side 2 of the Merlin 8 ProDOS diskette. Please note that *Sourceror* and *Sourceror.FP* are two entirely different programs. Sourceror is the disassembler in Merlin 8/16; Sourceror.FP is a separate program that produces the source listing of Applesoft BASIC. If you are looking for details on the disassembler, see the section on Sourceror.

Sourceror.FP works by scanning the resident copy of Applesoft present in your computer and generating text files called Aplsoft.A, Aplsoft.B, Aplsoft.C, and Aplsoft.D.

To conserve space, these files contain macros that are defined in another file on the disk entitled Applesoft.S. This file, when assembled using the PRTR command, will print out a nicely formatted disassembly of Applesoft, automatically bringing in and using the Apsoft files as necessary. Exact details on doing this are outlined below.

NOTE: This is not an official source listing from Apple Computer, Inc., but rather a product of the Author's own research and interpretation of the original Applesoft ROM. Apple Computer, Inc. was not in any way involved in the preparation of this data, nor was the final product reviewed for accuracy by that company. Use of the term Apple should not be construed to represent any endorsement, official or otherwise, by Apple Computer, Inc.

Additionally, Roger Wagner Publishing, Inc. makes no warranties concerning the accuracy or usability of this data. It is provided solely for the entertainment of users of the Merlin 8/16 assembler.

WARNING: Sourceror.FP and some temporary work files will be deleted when Sourceror.FP is BRUN. For this reason, you should make a backup copy of the SOURCEROR.FP side of the Merlin 8 disk before proceeding. Use the backup copy to make the Applesoft listing as explained next.

PRINTING THE APPLESOFT SOURCE LISTING

To get a printed source listing of Applesoft BASIC, follow these steps:

1. Start the Merlin 8 ProDOS diskette.
2. The Merlin 8 Full Screen Editor uses the same part of memory as Sourceror.FP. To use Sourceror.FP, you will have to temporarily deactivate the Full Screen Editor.

Press D for Disk Command.
3. At the Disk Command prompt type: BRUN /MERLIN.8/UTILITIES/REMOVE.ED and press Return.
4. Turn the disk over to the Sourceror.FP side.
5. At the Disk Command prompt type: BRUN /APPLESOFT/SOURCEROR.FP and press Return.

A prompt appears to remind you that files will be deleted if you proceed (see warning above). If you are using a backup copy of Sourceror.FP, press Y to continue.

6. When SOURCEROR.FP finishes, type L to LOAD a source file.
7. At the Load prompt type: /APPLESOFT/APPLESOFT and press Return.
8. To format the listing type: PRTR1""APPLESOFT LISTING and press Return.
9. Type ASM and press Return.

The assembler will prompt you to answer several questions about the format of the printout. You have the options of:

- a) PRINT THE DO OFF AREAS? (Y/N). Depending on whether you select the new ROM version (see next question), this choice gives you the option of printing those parts that are not actually assembled within the listing. Since the completed listing with XREF listings is on the order of 150 pages long, you probably won't want to print this listing too many times. We recommend answering Y to this first option.
- b) ASSEMBLE NEW ROM VERSION? (Y/N). If you answer N, you'll get the Apple II+ version of Applesoft. If you answer Y, you will be prompted with:
- c) ASSEMBLE IIC VERSION? (Y/N). If you have an Apple IIgs, IIc or Enhanced IIe, you will want to answer Y to this. If you want the original IIe version, answer N.

For each question, remember to just press the Y or N keys. You do not have to type "Yes" or "No."

In the example above, the PRTR command will send output to slot 1 and will print "APPLESOFT LISTING" as a header at the top of every page.

Merlin 8 will then ask "GIVE VALUE FOR SAVEOBJ :". This refers to whether or not you want to save object code generated by the assembly. It is recommended that you answer Ø. This is all you need to do to begin the printing process. If you answer 1 instead, you will save object code at the cost of slowing down the system. Saved object code allows you to verify it against where it was taken from.

Merlin 8 will now execute the first assembler pass. The disk will be accessed a few times, sometimes with long periods between accesses. This is normal. The entire first pass takes about 2 minutes.

Merlin 8 will then begin to print out a completely disassembled and commented listing of Applesoft. It will take about 105 pages including the symbol tables and nearly an hour and a half to print out at a printer rate of 80 characters per second.

APPLESOFT SOURCE CROSS REFERENCE LISTING

Although 105 pages of Applesoft source would seem like enough to keep one busy for at least a year, Merlin 8 also offers another source of Applesoft internal information - Applesoft internal address, subroutine and zero page cross references. By using the XREFA utility with the Applesoft source you can produce a listing of every subroutine, zero page address and where they are used and called. This is invaluable information for the programmer who desires to make use of the routines inside Applesoft in his own programs.

Assume, for example, that a user program is called by a running Applesoft program. Also assume that the programmer makes calls to some internal Applesoft routines and that the programmer wishes to use zero page locations \$50 and \$51 as temporary registers or pointers. This cross reference will immediately inform the programmer whether or not the routines that his program uses will destroy the contents of these two locations and cause difficult to find bugs in his program.

IMPORTANT: You *must* produce the source files before they can be cross-referenced. This is done by following the instructions on pages 192-194. Do this now if you have not already done so.

Steps to print an Applesoft cross reference:

1. Start the Merlin 8 ProDOS diskette.
2. Insert the Sourceror.FP diskette.
3. Press L to LOAD, then type: /APPLESOFT/APPLESOFT and press Return.
4. Type Q to quit the Editor, and return to the Main Menu.
5. Insert the Merlin 8 ProDOS diskette.
6. Press D for Disk command, then type: BRUN /MERLIN.8/UTILITIES/XREFA and press Return.
7. Enter the Editor, the type: PRTR1""APPLESOFT XREF and press Return.
8. Then type: USER 3 and press Return.
9. Type ASM and press Return.

You'll be prompted as follows:

Print DO OFF areas ? (Y/N)

You may answer Y or N. {See the previous section for an explanation of these questions...}

Assemble new ROM version ? (Y/N)

If you answer N, you'll get the Apple II+ version.
If you answer Y, you'll be prompted with:

Assemble //c version ? (Y/N)

Answer Y for the IIgs, IIc and Enhanced IIe version.
Answer N for the original IIe version.

- 11) Insert the Sourceror.FP diskette at the prompt:
"Insert /APPLESOFT/APLSOFT.A.S"
- 12) Press Return to start the cross-reference process.

The Applesoft source will again be assembled. This time, however, the XREFA program will limit your printed output to the cross reference table. Note that this process also takes quite a bit of time prior to printing.

UTILITY PROGRAMS

AUTO EDIT (MERLIN 16 ONLY)

This is a powerful utility but caution should be used with Auto Edit. It can be used from the Main Menu by typing BRUN UTIL/AUTO.EDIT and pressing Return. Another version called AUTO.EDIT.2 can also be loaded in a similar manner. It is more interactive and less dangerous but it is also much slower.

This utility creates an Open-Apple-\ command which is only accepted if a range is currently selected. This command lets you input an auto-edit string which will act on all lines of the selected range except comment lines. You can use ^ and an alpha character for control character commands. For example, the string ^U^U^D01 will move the cursor over twice, delete the next character and then insert the 01 characters, assuming the insert cursor is active.

Auto Edit 2 changes each line according to the edit string and then waits for a keypress. Pressing Escape, Control-C, or Control-X will cancel the entire process. Any non-control character such as Space will accept the change and move to the next line. Any control character such as Return will reject the change and move to the next line in the selected text.

CLASSIC DESK ACCESSORIES

Calendar - Notepad - Rational Calculator

The Merlin 16 disk contains commented source files for three Classic Desk Accessories. These are Calendar, Notepad, and Rational Calculator and the various files are located in the Library subdirectory. Please note these files are only offered as demonstrations of Classical Desk Accessories on the IIGs. You may use these files in your own programs but they *may not* be used in any program intended for commercial distribution. As noted in the source files for these programs, Glen Bredon reserves the commercial rights and copyrights for these and all related files.

CLOCK - MERLIN 8 (ProDOS only)

This Merlin 8 utility is an interrupt driven software clock designed for the IIC which lacks a clock to do the time stamping available in ProDOS. It requires the IIC because it uses the VBLINT interrupt provision. **THIS UTILITY SHOULD BE USED WITH CAUTION!** If it is overwritten, anything can happen and probably will. Press Reset to turn off interrupts. The source files are provided in the SOURCE directory on the Merlin 8 ProDOS version.

CONV.LNK.REL (MERLIN 8 ProDOS ONLY)

This makes the Merlin 8 REL files compatible with Apple's RLOAD and RBOOT programs. It will convert a Merlin 8 LNK file to Apple's REL format only if there are no externals. You can BRUN it from the Main Menu. If there is a source file in memory already, it will just return. Thus, enter NEW first in the Editor if necessary before using it. You will be prompted for the pathname of the file to be converted. The program will do the conversion and set up the converted file for the Merlin 8 object save command. The CONV.LNK.REL utility does not write anything to disk and does not delete or otherwise damage the original file.

You will be prompted for the pathname of the file you want to convert. The program will do the conversion and set up the converted file for the Merlin 8 object save command. The CONV.LNK.REL utility does not write anything to disk and does not delete or otherwise damage the original file.

CONVERTER: APW Source Files to Merlin 16 Format

Source files created by the APW (Apple Programmer's Workshop) and ORCA/M assemblers can be loaded directly into the Merlin 16 editor, but they usually require some editing before assembly. This is primarily due to differences in the pseudo-ops used by each assembler to define things such as data storage. For example, Merlin 16 can define a hex byte with the statement:

```
LABEL    HEX    FF
```

whereas APW would use:

```
LABEL    DC    I1 '$FF'
```

to achieve the same result.

If the file name of the APW source file does not have the ".S" suffix, remember to put the reverse slash (/) at the end of the name you enter at the Main Menu of Merlin 16. For example, if the source file you wanted to load was named FILE.SRC, you would enter FILE.SRC/ as the file to load.

For short files, you can load the APW source file into the Merlin 16 editor, and edit or use the search & replace (⌘E) or find (⌘F) functions to change any offending syntax in each line. In the case of the above example, you could use the (⌘E) command to replace DC I1' with HEX. Notice the space is included in both the search and replace strings. You will have to hold down the Open-Apple key when you press Return to start the search & replace to tell Merlin 16 to do the search on parts of words in the source file. In addition, in this particular case, you will want to go back through and remove the single quote from the end of the \$FF operand.

In addition, APW source files expand the space between each field in the source files with as many spaces as it takes to fill the gap. In Merlin 16, a single space character automatically tabs the text to the next field. Thus, source files in APW are up to 50% larger than the equivalent file in Merlin 16.

To remove the extra spaces in an APW source file, use the Command Box (⌘O) command FIXS to remove unneeded spaces in the file.

Finally, save the converted source file to disk under a new name. Merlin 16 will automatically add the ".S" suffix. For example, entering FILE as the name to save under would appear as FILE.S in the disk directory.

Using CONVERTER

Rather than manually editing an entire source file, it is usually easier to use a utility provided on the Merlin 16 disk called CONVERTER. This utility will automatically make many of the changes necessary to convert an APW source file to a form more acceptable to Merlin 16. Converter is an Applesoft BASIC program that can be run from either a file selector like ProSel, the DeskTop, etc., or by first setting the prefix to /MERLIN.16/UTILITIES at the BASIC prompt, and then typing RUN CONVERTER.

When the Converter runs, the first step is to place the disk containing the APW file to be converted into a disk drive. Then choose menu item #1, Select a file from disk.

The program will then present a list of all the on-line disk volumes. Press the number key or use the arrow keys to select the disk you want to look at, and press Return.

The program will then display all the files in the main directory of that disk. Subdirectories are indicated by a slash at the end of the filename, for example, UTILITIES/. Selecting a subdirectory will then list all the files in that subdirectory. You can back out of a given subdirectory by pressing the Escape key.

When you see the file that you want to convert, press the number key and press Return.

The program will then load the existing source file into memory, make the necessary changes, and then write a new file back to disk. The suffix ".S" will automatically be added to the new filename.

After the conversion, you may either select another file to change, or you can press Open-Apple-Escape to go back to the Main Menu. From there, you can either exit to BASIC, a program selector, or select another file to change.

Most Apple IIgs programs make extensive use of the Apple IIgs Toolset macros. At the beginning of a converted program you'll probably find an instruction similar to USE FILE.MACROS. This tells the assembler to use a library of macro definitions stored on the disk. Rather than try to convert a macro file for a given program, it is better to use another Merlin 16 utility, MACGEN, to create a new library of the macros needed for the program you're converting. The next section describes how to use MACGEN.

Using MACGEN

The Merlin 16 disk contains a complete set of Apple IIgs Toolset macro definitions, along with other macro libraries such as MACROS.816 and UTIL.MACS. The brute-force way of using these libraries is to add whatever lines are necessary to use the particular macros you need. For example:

```
USE /MERLIN.16/TOOL.MACROS/MEM.MACS
USE /MERLIN.16/TOOL.MACROS/MISC.MACS
etc.
```

The disadvantage to this approach is that the entire macro library is read into memory, leaving little remaining free memory for your actual program.

Usually you would write the program using whatever macro calls you wish, and then run the program utility MACGEN. MACGEN looks through an entire source file, and makes a list of every macro call. It then compares this with all the calls in any existing macro libraries, and creates a new and condensed macro file of just those definitions used in your program. MACGEN must be used when converting an APW source file in order to create the condensed macro file needed by a particular program.

When converting an APW source file, or when using MACGEN in your own program, here's what to do:

1) MACGEN will need to be installed in Merlin 16. This is done by typing:

```
~/MERLIN.16/UTILITIES/MACGEN
```

from the Main Menu.

2) Now load the source file to be scanned for macros. For an Apple IIgs, ProDOS 16 program, the first two actual instructions in your program, disregarding comment lines, should be:

```
REL
DSK  FILENAME.L
```

where FILENAME is the name of the final object file you wish to create, and FILENAME.L will be the intermediate relocatable link file. This LNK file will be used by the linker to create the final object file, presumably named FILENAME.

Following the REL and DSK instruction will be the USE FILENAME instruction that loads the condensed macro definition file, which would look something like this:

```
USE  FILE.MACROS
```

For now, this should be either deleted, or you can add a semi-colon (;) to temporarily make it a comment, since the file has not been created yet. After making these changes, save the file to disk.

3) The editor workspace must be empty for MACGEN to operate, so type NEW in the Command Box to erase the source listing. Then, go to the Command Box again and type:

```
MAC "FILENAME"
```

where FILENAME.S is the source file to be scanned.

4) The source file will be read from the current disk directory, and the Macro Generator menu will appear:

```
[S]earch directory for macros.  
[D]isplay unresolved macro names.  
[C]atalog directory.  
[Q]uit macro generator.
```

Select:

MACGEN has already made a list of all the macro calls in your program, and pressing D will display them. This step is not required, but it may be of interest to you. You will now tell MACGEN where the macro libraries are that you want reconciled with the source listing.

Press S to start the search. MACGEN will ask what directory you want it to search within for existing macro libraries. In most cases, you will probably enter:

```
/MERLIN.16/TOOL.MACROS
```

but this can be any directory that has macro definition files in it.

As the macro files are read and compared to the source file, each filename will be printed on the screen. When the directory is processed, the menu above will be repeated. Since you will probably want to use MACROS.816 also, press S again and type in:

```
/MERLIN.16/GEN.MACROS
```

At this point, all macro definitions that could be found are copied into a new source file in the Merlin 16 editor. Before quitting MACGEN, you may want to press D to display any macros definitions that were not found. You can either make a note of them to add later, or you can press S again to scan another directory for more macros. Note that unresolved macros may be due to UPPER/lower case use that is inconsistent with the actual definitions. For example, _TLStartup might not be resolved because the actual name is _TLStartUp. Setting the PARMS file so that Merlin 16 is case insensitive is one way to eliminate this source of unresolved macros.

When you quit MACGEN, you'll want to save the condensed macro file for use by the converted APW program. For example, for the program SAMPLE, you might want to save the macro file under the name SAMPLE.MACROS. This would be referenced in the source program with a line like this:

USE SAMPLE.MACROS

Once the macro file is created, you can load the source file, and put in the active USE instruction to include the macro file in your source, or remove the semi-colon you put in to temporarily make that instruction a comment.

Note #1: Upper/lower case of the filename is not important in Merlin 16 opcodes and filenames.

Note #2: The Merlin 16 pseudo-op USE cannot be used in a Put file. It can only be called from within the "master" calling program. If the program you are assembling has a USE macro library reference, try moving it to the master calling program. See the Merlin 8/16 manual regarding Put files.

Nested Macros: MACGEN looks within macros to see if other macros are called. Thus, if the macro _QDStartUp uses the macro Tool, the name Tool will be added to the list of unresolved macros. Usually, the macro will be found in other files, and so will be removed from the unresolved macro names before you even see it. However, if a macro definition is encountered in a library before it is referenced from within a later macro, you will have to make a second pass through that directory so that MACGEN will pick it up the second time. In practice, this is rarely needed.

Final Editing

There are certain further substitutions and editing that will need to be done to complete the conversion process. The easiest way to find the lines that need changes is to try to assemble the file (⌘A), and see where the errors occur. Remember that before you can use Linker.GS, you will have to re-install Linker.GS by typing "-/MERLIN.16/LINKER.GS" from the Main Menu, since MACGEN displaces Linker.GS when it is loaded.

Some APW functions are duplicated in macro functions included in the UTIL.MACS and MACROS.816 files. Other functions in APW may have no equivalents, and so are converted to comments by the Converter program. Following is a list of points to look for:

Comment/Line length: The Converter program automatically shortens lines to the 64 character limit of the Merlin editor. Any text that is too long is turned into a comment with an asterisk or a semi-colon, and is made into a new line. This may make title boxes in the listing look uneven.

Case Sensitivity: In its original condition, Merlin 16 is case sensitive in its handling of labels. That is, it considers the label NAME to be different from Name or name. Case sensitivity gives you more versatility in labels, for example, using the macro QUIT for ProDOS 16 and Quit for ProDOS 8. It also results in faster assembly speeds. APW, however, is not specifically case sensitive unless the CASE directive is used. In addition, many sample programs available for the Apple IIs written with APW use a wide variety of conflicting-case labels. For example, a single program may use pulllong, PULLLONG, Pulllong and PullLong in different parts of the program, all referring to the same macro.

When converting an APW source file, you may either manually edit any conflicting labels to be consistent, or you may alternatively change the PARMs file for Merlin 16 to make Merlin 16 case insensitive (about line #120 in PARMs.S).

Duplicate Labels: APW blocks groups of labels in each START/END segment and considers all labels in that segment to be local, unless declared "known" to another segment by the APW using pseudo-op. In Merlin 16, all labels are global unless preceded with a colon, as in :LOOP. In assembling a program, you may get "Duplicate Label" errors as Merlin 16 encounters labels that were used repeatedly in many different segments. To resolve these, you may either rename offending labels, i.e. LOOP1, LOOP2, etc., or you may put a colon in front of the offending label, i.e. :LOOP.

The main caution has to do with when a label is declared "known" to one segment, but may be local in all others. For example, suppose you have a program with three program segments, and the label LOOP was used in segments #2 and #3, but only the LOOP in segment #3 was known to segment #1 through APW's using pseudo-op. Since Converter turns it into a comment, you would have to pay particular attention to the preserved using instruction and to the corresponding START/END segment it referenced when editing the source listing. In this example, you might leave the label LOOP in segment #3 alone, and change LOOP in segment #2 to :LOOP. Note that references to local labels such as :LOOP cannot cross global labels. Thus, the following code segment would not work:

```

                BNE     :DONE
TEST2          CMP     #$7F
                BEQ     AGAIN
:DONE          RTS

```

Processor Mode: Merlin 8 and Merlin 16, as configured in the original package, start assembling a file assuming that the microprocessor is in the 8-bit mode (MX %11). In the case of Merlin 16, this is done to minimize errors when assembling source files written with earlier (8-bit) versions of Merlin, such as Merlin, Merlin Pro, Merlin 8 or Big Mac. When assembling programs to run under ProDOS 16, it is very important that you make sure the assembler starts assembling the source in the full 16-bit (MX %00) mode. There are two ways to handle this. The first is to just include the Merlin pseudo-op MX %00 on one of the first lines of each source file. You may alternatively use the macro M65816 1, as defined in the Macros.816 file, which accomplishes the same function. The Converter program automatically puts the proper instructions in when converting an APW program, so this is only a concern when manually converting a program, or writing a ProDOS 16 program from scratch.

The other choice is to permanently change the appropriate bit in the flag in the Merlin 16 PARMs file (approx. line #75) so that Merlin 16 always starts in the 16-bit mode. Important: If you do make this change, you will have to be careful to remember to use an MX %11 (8-bit mode) instruction in any of your older ProDOS 8, etc. source files before assembling them with the altered Merlin 16.

In addition, here are some of the APW pseudo-ops that are automatically handled by the Converter, but which may be of some interest:

65816 ON In APW, this enables the 65816 assembly mode. In Merlin 16, this is equivalent to two XC instructions, plus an MX %00. This is automatically added to converted APW source files by Converter, but there is also a macro named M65816 in MACROS.816 that can be used.

LONGI ON/OFF Merlin 16 uses 0 and 1 as substitutes for APW's ON and OFF operands. LONGI and LONGA have equivalent macros in MACROS.816.

START/END APW delimits each program segment within a listing with a START and END pseudo-op. All labels within a segment are considered local by APW except for the label that is associated with the START instruction. Local labels in APW within a segment may be made known to other segments with the APW "using label" instruction, which is not the same as Merlin 16's "using" instruction. The Converter program changes START to ENT so that the particular label on that line can be referenced by other Merlin 16 source files, if needed. END is converted to a comment, since END in Merlin 16 would signify the end of the entire source file. This structure may have to be considered when handling local/global labels and label conflicts.

SYMBOL ON/OFF In APW this determines whether the symbol table should be printed at the end of the source listing. In Merlin 16 this is controlled by the presence of LST ON/OFF at the end of the listing.

GEN ON/OFF In APW, GEN ON tells the assembler to print the expanded form of macros. EXPMAC is an equivalent function macro in Merlin 16 that duplicates this. Note: EXP is the actual Merlin 16 directive. See EXP in the Merlin 8/16 manual.

Programmer's Note: The Converter uses the file DICTIONARY in the UTILITIES subdirectory in part of the conversion process. The Dictionary file contains search & replace entries that translate APW opcodes to Merlin 16 equivalents. For example, the file Dictionary has the entry:

```
COPY#PUT
```

This tells the converter to replace the opcode COPY with PUT. Thus the APW source line:

```
COPY  HIRES.STUFF
```

would become:

```
PUT  HIRES.STUFF
```

The pound sign (#) is the delimiter between the search and replacement strings. Replacement strings may include spaces and semi-colons to move the new statements to the comment field, as is done with END becoming ;END. A pseudo-op can be removed completely by leaving the replacement field blank.

You can edit the Dictionary file with Merlin 16 to add new search & replace definitions. Remember to use the filename DICTIONARY/ when loading or saving. You will probably also want to turn off the formatting (tabbing to assembler fields) with the Command Box TAB command.

The Converter changes ON and OFF arguments to certain APW directives, such as absaddr, 65816, case, longa, longi, etc., to 1 and 0 so that the ON/OFF argument can be handled by an equivalent Merlin 16 macro. These macros are presently defined in the file MACROS.816 in the GEN.MACROS subdirectory on the Merlin.16 disk.

The source files for the Converter utility are also included on the Merlin.16 disk. Although we do not provide specific technical support for making alterations to this program, the source files are well-commented, and you may wish to make changes to the program for your own use.

Sample APW Program: A sample program, called SIMP.SRC, is included on the Merlin 16 disk in the directory SAMPLES.APW. You may want to try the procedure described here to practice converting an APW source file before tackling other files. A copy of the SIMP.SRC in converted Merlin format, and the final output file, SIMP, is included in the SAMPLES.M16 directory. SIMP is a ProDOS 16 S16 type file that can be launched from the Finder or DeskTop.

As a general exercise in assembling, linking and running a ProDOS 16 application that uses many Apple IIgs tools, there is also a program in the Samples.M16 directory called BRICKOUT. This file can be assembled, linked and saved by linking the command file GAME.CMD.S. This file was converted from a listing originally in the APW format.

CROSS REFERENCE PROGRAMS

Xref - Xrefa

These utilities provide a convenient means of generating a cross-reference listing of all labels used within a Merlin 8/16 source program.

Such a listing can help you quickly find, identify and trace values throughout a program. This becomes especially important when attempting to understand, debug or fine tune portions of code within a large program.

Merlin 8/16 provides a printout of its symbol table only at the end of a successful assembly, provided that you have not defeated this feature with the LST OFF pseudo op code. While the symbol table allows you to see what the actual value or address of a label is, it does not allow you to follow the use of the label through the program.

This is where the XREF programs come in.

XREF gives you a complete alphabetical and numerical printout of label usage within an assembly language program. XREFA gives a cross reference table by address. This is more useful for large sources containing lots of PUT files. It also does not use as much space for its cross-reference data and therefore can handle larger source files than XREF.

Sample Symbol Table Printout

Symbol table - alphabetical order:

```
ADD    =$F786    BC    =$F7B0    BK    =$F706
```

Symbol table - numerical order:

```
BK     =$F706    ADD    = $F786    BC     =$F7B0
```

Sample Xref Printout

Cross referenced symbol table - alphabetical order:

```
ADD    =$F786  101  185*
BC     =$F7B0  90   207*
BK     =$F706  104  121*
```

Cross referenced symbol table - numerical order:

```
BK     =$F706  104  121*
ADD    =$F786  101  185*
BC     =$F7B0  90   207*
```

As you can see from the above example, the definition or actual value of the label is indicated by the equal (=) sign, and the line number of each line in the source file that the label appears in is listed to the right of the definition. In addition, the line number where the label is either defined or used as a major entry point is suffixed or flagged with an asterisk (*).

An added feature is a special notation for additional source files that are brought in during assembly with the PUT pseudo opcode. For example, 134.82 indicates line number 134 of the main source file is the line containing the PUT opcode and line number 82 of the PUT file is where the label is actually used.

Using Xref

1. From the Main Menu, make sure you have saved the file and Merlin 8/16 is in the currently selected drive.
2. Merlin 16: Press D for Disk Command.
At the Disk Command prompt type BRUN /MERLIN.16/UTILITIES/XREF and press Return.

Merlin 8 ProDOS version: Press D for Disk Command.

At the Disk Command prompt type BRUN /MERLIN.8/UTIL/XREF and press Return.

Merlin 8 DOS 3.3 version: From the Main Menu press C to catalog the disk. At the Command prompt type BRUN XREF and press Return.

3. Enter the Editor, then type the appropriate USER command:

USER 0 - Print assembly listing and alphabetical cross reference only. USER has the same effect as USER 0.

USER 1 - Print assembly listing and both alphabetical and numerically sorted cross reference listings.

USER 2 - Do not print assembly listing but print alphabetical cross reference only.

USER 3 - Do not print assembly listing but print both alphabetical and numerical cross reference listings.

For example, to print a cross-reference listing only to your printer, you could type in:

```
PRTR 1""  
USER 3  
ASM
```

USER commands 0-3 cause labels within conditional assembly areas with the DO condition OFF to be ignored and not printed in the cross reference table.

You can also use USER commands 4-7 which are identical to USER 0-3 except that they cause labels within conditional assembly areas to be printed regardless of the DO setting. The only exception to this is that labels defined in such areas and not elsewhere will be ignored.

NOTE: You may change the USER command as many times as you wish, i.e. from USER 1 to USER 2 and so on. The change is not permanent until you enter the ASM command.

4. Merlin 16: Press Open-Apple-A to begin the assembly and printing process.

Merlin 8: Type ASM and press Return to begin the assembly and printing process.

Since the XREF programs require assembler output, code in areas with LST OFF will not be processed and labels in those areas will not appear in the table. In particular, it is essential to the proper working of XREF that the LST condition be ON at the end of assembly since the program also intercepts the regular symbol table output. For the same reason, the Control-D flush command must not be used during assembly. The program attempts to determine when the assembler is sending an error message on the first pass and it aborts assembly in this case, but this is not 100% reliable.

NOTE: When using macros with XREF, labels defined within macro definitions have no global meaning and therefore are not cross-referenced.

```

DEF      MAC                <---Macro definition
        CMP    #]1
        BNE   DONE
        ASL
DONE     <<<
-----
        DEF   GLOBAL        <---Beg. of program
                                <---Macro call

```

In the above example, variable GLOBAL will be cross referenced, but local label DONE will not.

XREFA Notes

This is an *address* cross reference program and is handy when you have lots of PUT files. Since this program needs only four bytes per cross reference instead of six, it can handle considerably larger sources. The "where defined" reference is not given here because it would equal the value of the label. The exception is EQUated labels where it would indicate the address counter when the equate is done. This also saves considerable space in the table for a larger source.

FORMATTER

This program is provided to enhance the use of Merlin 8/16 as a general text editor. It will automatically format a file into paragraphs using a specified line length. Paragraphs are separated by empty lines in the original file.

To use Formatter, you should first BRUN it from Main Menu. Formatter will then load itself into high memory.

This will simply set up the editor's USER vector. To format a file which is in memory, issue the USER command from the editor.

The formatter program will request a range to format. If you just specify one number, the file will be formatted from that line to the end. Then you will be asked for a line length, which must be less than 250. Finally, you may specify whether you want the file justified on both sides, rather than just on the left.

The first thing done by the program is to check whether or not each line of the file starts with a space. If not, a space is inserted at the start of each line. This is to be used to give a left margin using the editor's TAB command before using the PRINT command to print out the file.

Formatter uses inverse spaces for the fill required by two-sided justification. This is done so that they can be located and removed if you want to reformat the file later. It is important that you do not use the FIX or TEXT commands on a file after it has been formatted unless another copy has been saved. For files coming from external sources, it is desirable to first use the FIX command on them to make sure

they have the form expected by Formatter. For the same reason, it is advisable to reformat a file using only left justification prior to any edit of the file.

Don't forget to use the TABS command before printing out a formatted file.

KEYBOARD MACRO FILES

Edmac - Keymac

Edmac and Keymac are keyboard macro files for use with the Merlin 8/16 Editors. Edmac is for use with Merlin 8 and Merlin 16 full screen editors. Keymac is for the Merlin 8 line editor. A macro definition lets you type one key, and get a string of characters on the screen. This should not be confused with the assembler macros that Merlin 8/16 also supports.

An assembler macro is a definition of a set of assembler instructions, usually with variables, that you define within a given source listing. When the program is assembled, the assembler replaces the macro call with the series of lines that have been assigned to that macro.

A keyboard macro is only a substitute for a small amount of typing that you might do while you're in the editor itself.

For example, you've probably typed LDA many times in assembly language programs. With Edmac or Keymac installed, you could type Solid-Apple-3 (Option-3 on the Apple IIgs) and the characters LDA # would appear on the screen.

To install Edmac or Keymac, just BRUN it from the Main Menu, then enter the Editor and type USER from the Merlin 16 Command Box or Merlin 8 Command Mode. When you type one of the Edmac or Keymac Solid-Apple (Option- on the GS) commands shown on the following pages, the corresponding text will be inserted.

KEYBOARD MACRO EQUIVALENT CHART

What You Type:	To Get:	Comments
⌘ A	1 AND	
⌘ B	2 BVC	
⌘ C	3 CMP	
⌘ D	4 DFB	
⌘ E	5 EOR	
⌘ F	N/A	Keyboard Macro Not Assigned...
⌘ G	N/A	Keyboard Macro Not Assigned...
⌘ H	6 HEX	
⌘ I	N/A	Keyboard Macro Not Assigned...
⌘ J	7 JSR	
⌘ K	8 JMP	
⌘ L	9 LDA	
⌘ M	10 BMI	
⌘ N	11 BNE	
⌘ O	12 ORA	
⌘ P	13 BPL	
⌘ Q	14 BEQ	
⌘ R	15 RTS	; PLUS CARRIAGE RETURN
⌘ S	16 STA	
⌘ T	N/A	Keyboard Macro Not Assigned...
⌘ U	N/A	Keyboard Macro Not Assigned...
⌘ V	17 BVS	
⌘ W	N/A	Keyboard Macro Not Assigned...
⌘ X	18 LDX #	
⌘ Y	19 LDY #0	; LEAVES IN INSERT MODE
⌘ Z	20 LD #0	; LEAVES IN INSERT MODE
⌘ 1	N/A	Keyboard Macro Not Assigned...
⌘ 2	N/A	Keyboard Macro Not Assigned...
⌘ 3	21 LDA #	
⌘ 4	22 LABEL = \$; FOR HEX EQUATES
⌘ 5	23 LABEL DFB %	; FOR BINARY EQUATES
⌘ 6	24 N/A	Keyboard Macro Not Assigned...
⌘ 7	25 " & \$9F	; USED TO DEFINE A CONTROL CHAR.
⌘ 8	26 *-----	
⌘ 9	27 (), Y	; WITH INSERT MODE ON...
⌘ 0	28 (, X)	; WITH INSERT MODE ON...
⌘ ^J (down arrow)	29 PLA	; SAVE A, X, Y ON STACK
	30 TAY	; ALL 5 LINES WITH ONE MACRO.
	31 PLA	; (MACRO KEY = DOWN ARROW)
	32 TAX	
	33 PLA	
⌘ ^K (up arrow)	34 PHA	; RETRIEVE A, Y, Y FROM STACK
	35 TXA	; ALL 5 LINES WITH ONE MACRO.
	36 PHA	; (MACRO KEY = UP ARROW)
	37 TYA	
	38 PHA	

Ⓜ '	39	ASC ''	; SINGLE QUOTE WITH CURSOR BETWEEN QUOTES
Ⓜ "	40	ASC ""	; DOUBLE QUOTE WITH CURSOR BETWEEN QUOTES
Ⓜ \	41	ERR \	; FOR 'ERR' CHECK VALUE
Ⓜ .	42	MAC	; DEFINE A MACRO.
Ⓜ ,	43	EOM	; FINISH DEFINITION.
Ⓜ =	44	ADC	; = (SAME KEY AS '+')
Ⓜ -	45	SBC	; "-" FOR SUBTRACTION.
Ⓜ]	46]LOOP	; VARIABLE FOR A LOOP.
Ⓜ [47	[]	; INDIRECT LONG ADDRESSING MODE.

Macros can be added or edited by adding or changing the data statements in KEYMAC.S or EDMAC.S. The only requirement for each definition is that it begin with the macro key itself, and that the assigned string end with a "high bit off" (value < \$80) ASCII character.

MAKE DUMP

This Merlin 8/16 program will make a hex dump text file from an object file after a valid assembly of the source. Before using Make Dump, you should have already assembled and saved the source code. This is important because the existing source code in memory is destroyed when Make Dump is used. To use Make Dump, BRUN it from the Main Menu. Then load the desired source file and assemble it. Finally, type USER from the Command Box or Command Mode of the Editor.

The current source in memory will be replaced with a hex dump of the previously assembled object file. The last line of the file will contain a BSAVE with the correct address and length but *without a filename*. You can use the Editor to insert the desired filename. You can then save the file for use as an EXEC file. If you forget to insert the filename, Merlin 8/16 will give you a SYNTAX ERROR when the file is EXECed.

PRINTFILER

Printfiler is a utility designed to save an assembled listing to disk as a sequential text file. It optionally allows you to also select "file packing" for smaller space requirements and allows you to turn video output off for faster operation.

Text files generated by Printfiler can include the object code portion of a disassembled listing, something not normally available when saving a source file. This allows a complete display of an assembly language program and provides the convenience of not having to assemble the program to see the object code.

Printfiler can also be used to save a source listing to disk as a text file with the high bit clear (Merlin source files normally have the high bit set), and with the option of each tab field filled with spaces for compatibility with the APW (ORCA) assembler.

Printfiler Applications

Other examples of where Printfiler might be used include:

- Incorporating the assembled text file in a document being prepared by a word processor.
- Sending the file over a telephone line using a modem.
- Mailing the file to someone such as a magazine editor who wants to work with the complete disassembly without having to assemble the program.

Using Printfiler

1. Make sure that you've saved the source file before using Printfiler.

Merlin 16 version: Press D for Disk Command.

At the Command prompt type BRUN /MERLIN.16/UTILITIES/PRINTFILER and press Return.

Merlin 8 ProDOS version: Press D for Disk Command.

At the Command prompt type BRUN /MERLIN.8/UTILITIES/PRINTFILER and press Return.

Merlin 8 DOS 3.3 version: Press C to catalog the Merlin 8 disk.

At the Command prompt type BRUN PRINTFILER and press Return.

2. Press L to load the desired source file.
3. From the Main Menu select the drive to save the assembly to if necessary.

Merlin 16 version: Enter the Editor and press Open-Apple-O to open the Command Box.

From the Command Box type USER "FILENAME" where FILENAME is the name of your file. You may also use the PRTR command if you wish page headers to be sent with your listing. In that case enter the following instead of the USER command: PRTR 1 "T.MYFILE" MY PAGE HEADER and press Return.

Merlin 8 version: Enter the Editor by pressing E from the Main Menu. From the Command Mode prompt (:) type USER "FILENAME" where FILENAME is the name of your file. You may also use the PRTR command if you wish page headers to be sent with your listing. In that case enter the following instead of the USER command: PRTR 1 "T.MYFILE" MY PAGE HEADER and press Return.

There will be a short disk access and the name of the file used for the text save will appear on the screen.

4. Merlin 16 version: Press Open-Apple-A to begin the assembly and Printfiler will automatically assemble the source file directly to disk. Note that you will not see anything on your video screen because Printfiler is preconfigured to operate with the video output turned off for faster operation.

Merlin 8 version: From the Command Mode prompt type ASM and Printfiler will automatically assemble the source file directly to disk. Note that you will not see anything on your video screen because Printfiler is preconfigured to operate with the video output turned off for faster operation.

NOTE: You should not use a filename that already exists on the disk as an object file for the output file. Printfiler uses the exact filename with no prefix or suffix that you enter. Thus, if you had the source file PROGRAM.S and the object file PROGRAM on a disk, you would not want to use the name PROGRAM for the PRINTFILE name. PROGRAM.TEXT would be the recommended name.

Printfiler will also print the output from the Merlin 16 "L" (list with line numbers) and "P" (Print without line numbers) commands. To output a text file for the APW (ORCA) assembler with each tab field automatically filled with spaces, just use the "P" command in the command box after opening the text file to print the source file to a formatted text file with the high bit clear. This file can then be directly loaded by APW.

Changing Printfiler Options

Printfiler has two options that you may change: file packing and video output or echoing. In addition, you can make the change temporary or permanent.

File packing reduces the size of the text file saved to disk by replacing blanks from the source file with a single character with its high bit turned off. A listing of a packed file will display the packed blank characters as an inverse letter. Thus inverse A=1 blank, inverse B=2 blanks, inverse C=3 blanks, etc.

Unpacking means restoring the text file to its original appearance. Note that while you cannot assemble such a file, you can at least read it.

Echoing means printing on the screen what is sent to the disk. The time it takes to do this can slow Printfiler down.

The process of turning off video output or echoing makes Printfiler run approximately 25% faster. Additional speed can be gained by using packed files.

In addition, unpacked files are nearly twice as large as packed files and nearly three times the size of the original source file.

Changing Printfiler Options (temporarily)

To temporarily change the Printfiler options you can go to the Monitor from the Merlin 8/16 Editor with the MON command. Enter the following:

```
300:00 00   for packed, video off, or.  
300:00 80   for packed, video on, or  
300:80 00   for unpacked, video off, or  
300:80 80   for unpacked, video on
```

The Printfiler standard values are 300:80 00 (unpacked, video off)

Press Return, then Control-Y, then Return again to go back to Merlin 8/16. The values you select will stay in effect until you BRUN PRINTFILER again.

Changing Printfiler Options (permanently)

1. Load Printfiler as above and assemble with it. During assembly, it will prompt you with the following questions:
2. GIVE VALUE FOR FORMAT
Press 0 to turn on the Pack option.. Press 1 to turn off the Pack option.
3. GIVE VALUE FOR MONITOR
Press 0 to turn video output or echoing off. Press 1 to turn echoing on.

Printfiler will then immediately assemble into object code.

4. Quit the editor and save the Object code. Any time you BRUN this object code, it will use the values you specified in steps 2 and 3. Thus, it is possible to use different versions of Printfiler instead of setting options.

TXTED (MERLIN 16 ONLY)

This is an alternate version of the Merlin 16 Full Screen Editor which will break a line upon a carriage return, and carriage returns are deletable. TXTED can be substituted from the Main Menu by typing BRUN UTIL/TXTED and pressing Return.

TYPE.CHANGER

This utility will change the ProDOS file type of any file on a disk. It is an Applesoft BASIC program that can be run from either a file selector like ProSel, the DeskTop, etc., or by first setting the prefix to /MERLIN.16/UTILITIES at the BASIC prompt, and then typing RUN TYPE.CHANGER.

When Type.Changer runs, the first step is to place the disk containing the file whose type is to be changed into a disk drive. Then do the following:

- 1) Choose menu item #1, Select a file from disk.
- 2) The program will then present a list of all the on-line disk volumes. Press the number or use the arrow keys to select the disk you want to look at, and press Return.
- 3) The program will then display all the files in the main directory of that disk. Subdirectories are indicated by a slash at the end of the filename, i.e. UTILITIES/. Selecting a subdirectory will then list all the files in that subdirectory. You can back out of a given subdirectory by pressing the Escape key.
- 4) When you see the file whose type you want to change, press the number key and press Return.
- 5) The program will then display the current filetype of that file showing both abbreviated file code and hex filetype. For example, an Applesoft BASIC file would be listed with a file code BAS and a filetype of \$FC.
- 6) You can enter the new filetype you want as either the "official" file code, such as BIN, S16, etc., or the hex code, such as \$FF. If entering the hex code, you must include the dollar sign at the beginning of the number.
- 7) After the change, the new file code and type will be displayed. Pressing a key will return to the Main Menu. At that point you can either exit to BASIC, a program selector, or select another file to change.

NOTE: The file codes and types are all contained in DATA statements within the Type.Changer program. These may be added to or edited as you desire to support new and/or custom file types.

ADDITIONAL MERLIN 8/16 RESOURCE FILES

The Merlin 8/16 disks contain a number of source files for your reference including PUT file examples, general macros that can be used in any program, Classic Desk Accessories, and Toolbox macros for the IIs. These files are provided as resource material and can be helpful in understanding how these tools are designed and implemented.

INDEX

"!", Exclusive OR 81
 "&", Logical AND 81
 "/" (, - , .), in Macros 131
 ". ,",
 as a logical OR 77, 81
 "/",
 to abort a Change 38
 Division operator 81
 Editor - to List from last line 46, 69, 72
 Editor - to abort List 42
 in file names 18
 to abort a Find 39
 Line Range Delimiter 36, 37, 61, 62
 in Macros 131
 Sourceror (Cancel) 189
 * for Comments 5, 77, 78
 . (period),
 Listings 46, 69, 72
 in Macros 131
 256 Externals 183

A

A:Append File 19
 About the Assembler Documentation 75
 About the Command Mode Documentation 36
 About the Command Box Documentation 61
 About the Linker Documentation 138
 Absolute addresses, and Linker 140, 144, 151
 Accumulator mode addressing 83-85
 Add/Insert Mode Editing Commands 5-7, 8-9,
 37, 63, 72
 Addition operation, in operands 81
 Addressing Modes 83-85
 ADR 111
 Linker command 152
 ADRL 111
 ALI 159
 All text, to select 29, 34, 57, 71
 AND operation, in operands 81-82
 Angle brackets in Documentation 36, 61
 Append a file 19

Apple Keys (⌘)

⌘A 9, 52, 71
 ⌘B 27, 34, 52, 71
 ⌘C 52, 71
 ⌘D 27, 34, 53, 71
 ⌘E 27, 34, 53, 71
 ⌘F 28, 34, 54, 71
 ⌘H 54, 71
 ⌘I 28, 34, 55, 71
 ⌘L 28, 34, 55, 71
 ⌘N 29, 34, 55, 71
 ⌘O 8, 55, 71
 ⌘Q 8, 12, 29, 34, 55, 58, 71
 ⌘R 29, 34, 56, 71
 ⌘T 30, 34, 56, 71
 ⌘V 30, 34, 56, 71
 ⌘W 30, 34, 56, 71
 ⌘X 30, 34, 57, 71
 ⌘Y 31, 34, 57, 71
 ⌘Z 30, 34, 57, 71
 ⌘ Up 31, 34, 58, 71
 ⌘ Down 31, 34, 58, 71
 ⌘ Right 71
 ⌘ DEL 31, 34, 57, 71
 ⌘ TAB 31, 34, 55, 58, 71
 ⌘ - 31, 34, 60, 71
 ⌘ = 31, 34, 60, 71
 ⌘1 58, 71
 ⌘2 59, 71
 ⌘3 59, 71
 ⌘4 59, 71
 ⌘6 59, 71
 ⌘8 32, 34, 59, 71
 ⌘9 32, 34, 71
 ⌘ 60, 71

Applesoft source listing 192
 APW to Merlin Converter 198-205
 Arithmetic and Logical Expressions 81-82
 ASC 107

- Assembler,
 - to Pause 121, 123
 - Pseudo Opcode Descriptions 87
 - Syntax Conventions 77, 83
- Assembling large files, and PUT, SAV, DSK 93-96, 98-99, 100-101, 147-148, 152, 153, 155-157
- Assembly 9-11, 47, 52, 63, 73
- ASM,
 - Command 9-11, 47, 63, 72
 - and PRTR 43-44, 47, 66, 72
 - Linker command file 151, 152
- AST 103
- Asterisks (*),
 - Comments 5, 77, 78
 - Line of in a comment 31, 34, 59, 71, 103
- Auto edit 73, 197
- B**
- Back-up copies of Merlin 3
- Backing up Program Counter (DS) 89, 112
- Backwards DELETE, in EDIT mode 26, 33, 50, 70 (see also Control-D)
- Bad,
 - Address mode 179
 - Branch 179
 - External 179
 - Input 179
 - Label 179
 - OBJ 179
 - Opcode 179
 - ORG 179
 - PUT 180
 - REL 180
 - SAV 180
 - Variable 180
- Beginning, move to,
 - of line of text 24, 33, 48
 - of source file 27, 34, 52
- Bell, turning off 167, 172
- BINARY files 12, 13, 17
- Binary numbers 80
- BGE opcode 86
- BLOAD 14, 19, 20, 21, 89
- Block cursor 24, 26, 49, 167
- BLT opcode 86
- Branching,
 - to Variables and Local Labels 79, 86
- BRK 180
- BRUN 19, 20, 21
- BSAVE 19
- Bugs, common cause of 87
- Building Expressions 81-82
- C**
- C: Catalog 17-18
 - pause 17-18
- CALL 14
- Case sensitive labels 88
- Center screen 30, 57
- Change 38
- Change drive 19
- Change ProDOS filetype 214-215
- Change Word 39
- Changing Printfiler's Options 213-214
- Changing Sourceror's Label Table 191
- Character case change 24, 33, 49, 62
- Character insert mode 24, 26, 33, 49, 70
- Characters per line 34, 63, 73, 77, 167, 170, 172
- Checksum, in object code 119-120
- CHK 119-120
- Classic Desk Accessories 197
- Clipboard 30, 34, 52, 73
- Clock 197
- Command box 8, 55, 61
- Command mode (Merlin 8) 15, 34, 36
- Command summary 33-34, 70-72
- Comments 5, 77
- Comment length 34, 63, 73
- Conditional Assembly 82-83, 116-118, 151, 152

- Conditional Pseudo Ops 114-118
 - Configuration 4, 72, 166
 - Control characters 25, 33, 49, 70
 - Control-B (go to line begin) 24, 33, 70
 - Control-C,
 - during Catalog Command 17
 - to abort assembly 33, 47, 48, 70, 73
 - to abort List 42
 - to abort a Change 38
 - to abort a Find 39
 - or Control-X (to cancel lines) 34
 - Control-D (delete) 9, 24, 33, 49, 70
 - LST ON/OFF 47
 - Control-F (find) 24, 33, 49, 70
 - Control-I (insert) 24, 26, 33, 49, 70
 - Control-L 24, 33, 49, 70
 - Control-N (go to line end) 25, 33, 49, 70
 - Control-O (insert "other" character) 25, 33, 49, 70
 - Control-R (restore line) 25, 33, 70
 - Control-S (status box) 25, 50, 70
 - Control-T 25, 33, 50, 70
 - Control-W 25, 33, 50, 70
 - Control-X (to Cancel global exchange) 25, 33, 50, 70
 - Control-Y 25, 33, 50, 70
 - Control-Y, to return to Main Menu 15
 - Conv.lnk.rel 198
 - Converter, APW to Merlin source 198-205
 - Copy,
 - Command Mode, Editor 38
 - Full Screen Editor 52, 71
 - Copying Merlin 3
 - Cursor keys 26, 31
 - Cursors,
 - appearance/types 24, 26, 49, 70, 167
 - moving 26, 33, 50
 - Customizing Merlin 4, 72, 166
 - Cut 30, 33, 56, 57
 - CW (Change Word) 39
 - CYC 103-104
 - Cycle times, 103-104
 - column to print 167, 172
- ## D
- Disk Commands 19
 - D:Drive Change 19
 - DA 110
 - DAT 104, 153
 - Data,
 - definition of 76
 - immediate 76, 83
 - string 74, 107-109
 - storage 110-113
 - tables in programs 113
 - Data and Storage Allocation Pseudo Ops 76
 - Date, to set 22
 - DB 110
 - DCI 108
 - DDB 110
 - Decimal numbers 80
 - Defining,
 - a keyboard macro 209-211
 - a local label 78-79
 - a local variable 79
 - a Macro 127, 128-132, 133-134
 - Delete,
 - characters in a line 9, 24, 26, 33, 49, 50, 70
 - lines in Editor Command Mode 39
 - lines in Full Screen Editor 9, 27, 52
 - to end of line 25, 33, 50
 - entire Source File (New) 21, 43, 65
 - files on disk 19
 - Delete key 26, 33, 50, 70
 - Delimited Strings, 37, 61, 76
 - as an operand 76
 - DEND 102
 - DFB 110
 - Dictionary full 180
 - Disassembling,
 - raw object code (Sourceror) 184-191

Disk files,
 names - see "filenames"
 renaming 19
Division operation, in operands 81
DO 114
DOS 3.3 technical information 164
Drive change 19
DS 90, 112
 and Linker 112, 139, 143, 159
DSK 90, 100
 and the Linker 147, 149, 155-156
DUM 101
Duplicate symbol 146, 180
Duplicating Merlin disk 3
DW 110

E

Edit command 38
Edit Mode Commands 33-34, 70-72
Editor 5-8, 21
Edmac 35, 73, 209-211
ELSE 114
END 101, 153
End of line,
 marker 23, 48
 move to 25, 33, 49
End of source, move to 29, 34, 55
ENT 88, 141, 142, 153
EOM or <<< 127
EQU (=) 87
ERR 120
 and Linker 120-121, 139, 143
ERR: Memory Full 181
Error Messages (general) 179-183
Escape 26, 33, 50, 70
Evaluation of expressions 45, 69, 72, 81
Exchange, global 27, 34, 53
EXCLUSIVE OR operation, in operands 81-82
EXP ON/OFF/ONLY 104
EXT 141, 153
Expressions Allowed by the Assembler 81-83

F

Fast keys 50
Filenames,
 ASM 151
 DSK 90, 91, 100-101, 147, 155-156, 161
 LIB 154
 Link 144
 Linker command files 154-155
 Linker name files 144-145, 148-149
 LNK 147, 152, 155-156
 Multiple LNK files 160
 object 12, 13, 14, 19, 2
 OVR 152
 PUT 77, 93-96, 152
 Quick Link 159-160
 REL 147, 155-156, 161
 SAV 90, 92, 98-99, 153
 source 12, 13, 18, 19, 21, 22, 77
 USE 77, 97, 136
Filetype mismatch error 146, 180
FIN 115
Find,
 a character 24, 33, 49, 70
 a string 25, 28, 39, 54, 71
 a word 25, 28, 30, 39, 40, 50, 54, 56
Find and Replace (global exchange) 25, 27, 34,
 38, 53
FIX 40, 63, 72
FLS 108
Forced Assembly Errors 120, 121, 143
Formfeed, printer 166, 170
FW (Find Word) 40
Format error 180
Formatter 208-209
Formatting Pseudo Ops 103-106
Full Screen Editor,
 commands 8-9
 Merlin 8 23-35
 Merlin 16 48-60
 entering 5, 21, 23, 48
 quitting 7, 44, 66, 71

G

GET 15, 41, 64, 72
GET Command, and Linked files 146
Global Exchange 25, 27, 34, 38, 53
GS Linker 158, 161-162, 163

H

Half-screen editing 54
H (Hex) 187
Hard disks 170
HEX 111
Hex-Dec Conversion 41, 64
Hex data 80, 107, 110

I

IF 115
Illegal,
 char in operand 180
 file type 181
 forward reference 181
 relative adrs 181
Initialization string, for printer 43, 66, 175
Insert,
 with TAB key 24, 26, 28, 49, 50
 character mode 9, 26, 49, 50
 lines 8, 26, 28, 55, 58
 control characters 25, 49
Integer division, in operands 81
Inverse spaces 44, 67
Immediate Data Syntax 80, 83
Immediate Data vs. Addresses 80
INV 108

K

KBD 121
Keyboard input during assembly 121
Keymac 35, 209-211
KND 159

L

L (list - Sourceror) 187
L: Load Source 18
Labels,
 proper form of 6, 77
 length 77
 tables, changing Sourceror's 191
 case sensitivity 88
 jump to 28, 34, 55, 70
LENgth 42
LIB 154
Line length 34, 73, 78
Line numbers,
 in Command box 62
 in Command mode 36
Lines of text,
 to delete 9, 27, 31, 34, 39, 52, 57, 70
 to insert 9, 18, 28, 33, 34, 55, 70
 to replace 25, 29, 30, 34, 50, 56, 70
Lines per page 166, 170
Link 72, 139
Linker 137-163
 and DS opcode 139, 142
 and DSK opcode 147-149
 and ENT opcode 138, 142
 and ERR opcode 139, 143
 and EXT opcode 138, 141
 and ORG opcode 144, 152
 and REL opcode 138, 140-141
 and SAV opcode 153
 File Names 144-145
Linking process 146
List 8, 41, 64, 72
 and PRTR 43
 from last listed line 46, 69, 72
 from last specified line 46, 72
 to printer, formatted 41, 64, 72
 to screen, formatted 41, 64, 72
 without line numbers 43, 66, 72
 to abort 41
 to pause 41

- Listing,
 - CYCLE times 103-104, 167, 172
 - DO OFF 114, 167, 172
 - Macros 104
 - limiting bytes printed in 167, 172
- LKV 154
- LNK 152
- Load a source file 18
- Local Labels, Global Labels & Variables 78-80
- Local Variables,
 - defining 79-80
 - and PUT files 96
- Locate a label or line 28, 34, 55
- Lock file on disk 19
- Logical operations, in operands 81
- Lower case/upper case,
 - to change 24
 - in labels 88
- LST ON/OFF 105
- LSTDO or LSTDO OFF 105
- LSTDO, configuring 167, 172
- LUP 122
- M
- MAC 127
- Macgen (utility) 200-202
- Macros 127-136
 - defining 127, 128-132, 133-134
 - and PUT files 96
 - libraries, and USE opcode 97, 136
 - libraries, provided with Merlin 136
 - listings 104
 - nested 134
 - Pseudo Ops for, 127-129
 - variables 129-133
- Main menu 5, 17, 55
- Making Back-up Copies of Merlin 3
- Make dump 211
- Markers 28-29, 55, 56
- Maximum,
 - length of comments 34, 63, 73
 - length of labels 77
- Memory,
 - IN USE 146, 182
 - full 181
 - full error, in Sourceror 191
 - status 25, 33, 50, 70
 - use by Merlin 166
- Merlin,
 - Memory Map 176-178
 - internal entry points 124
- Misalignment error 182
- Miscellaneous Pseudo Ops 119-126
- Mistakes, fixing 25, 29, 34 (see also Control-R and ⌘R)
- MONitor 42, 65, 72
- MOVE 42 (see Cut, Copy & Paste)
- Moving,
 - the cursor 26, 33, 48, 49, 50
 - text 29, 42
- Multiple input LNK files 160
- Multiple output files 157
- Multiplication operation, in operands 81
- MX 84, 123
- N
- N (Normal - Sourceror) 187
- Nesting error 182
- Nested Macros 134
- New 21, 43, 65, 72
- Not macro error 182
- Not resolved error 146
- Number Format (Binary, Decimal, Hex) 80
- O
- O: Object Save Command 12, 21
 - command, and Linker 146, 153
- O (ORG - Sourceror) 187
- OBJ 92
- Object code 13, 21
- Online command 19, 20

- Opcode and Pseudo Opcode Conventions 78
Operand and Comment Length Conventions 78
OR operation, in operands 81-82
ORG 6, 21, 89
 and the Linker 91, 152
Out of Memory 21, 182
- P**
- PAG 106
Page Header, in listing 106
Parentheses,
 and Precedence in Expressions 82
PARMS file 4, 34, 55, 72, 170
Paste 30, 34, 56
Pathnames 77
PAU 123
Personalizing Merlin 4, 72, 166
PFX 19, 20
PMC or >>> 128
Precedence, in operand expressions 82
Prefix, ProDOS 13, 19, 20
Preliminary Definitions 75-76
Primitive expressions 81
Print command,
 and PRTR 43, 59, 66
 without line numbers 43, 66
Printer,
 slot#, in PRTR command 43, 66
 string, in PRTR command 43, 175
 to print Catalog 17
Printfiler 211-214
PRinTeR 43, 59, 66, 175 (see also TTL, pg
 106; PAG, pg 106)
ProDOS technical information 168
PUT 93-96, 137
 linker command 152
- Q**
- Q (Quit),
 to BASIC or program selector 22, 66
 to Main Menu 7, 44, 72
 Sourceror 187
Quick Link 159
- R**
- R: Read Text File 21
R (Read - Sourceror) 188
Range Error 183
REL Files 91, 92, 140
 and the ERR Pseudo Op 143
 and the Linker 112
 and OBJ opcode 91
Relative expressions, and Linker 140
Renaming disk files 19
Replace 29, 56
Restoring lines in Edit mode 25
Return key 18, 33, 70
REV 108
Reversed string data 108
RTS return to Merlin 6
Running a program 14
- S**
- S: Save source file 10, 13
SAV 90, 98, 153
".S",
 suffix in file names 10, 18, 96, 97
Select all text 29
Semicolon,
 for comments 6
 in macros 131
SKP 106
Slot 19, 20
Source code format 7, 77-78
Sourceror,
 Merlin 8 184-191
 Merlin 16 189
 XL version 189
Spaces in a text line 7
Special macro variables 129-133

- Split screen editing 54
- Status, memory 25, 33, 50, 70
- STR 109
- String Delimiters 37, 61
- Suggested reading 2-3
- SW 123
- Sweet 16 85, 123
- SYM 67, 72
- Symbol table, description of 10-11, 165
- Syntax error 183
- Syntax, Source code 7, 75, 77
- System requirements 1

- T

- T (Text - Sourceror) 188
- TAB Key 18, 24, 26, 33, 50, 70
- Tabs,
 - to set Tabs 44, 67, 72, 167, 172
 - to zero Tabs 21, 67, 72
 - and word processing text files 44, 67
- Technical Information 164-178
- TEXT 44, 67, 172
- Text files 17, 21
- Text, select all 29, 57
- TR ON/OFF/ADR 106
- TRuncOff 44, 68, 72
- TRuncON 45, 68, 72
- TTL 106 (see PRTR also)
- Two Externals 183
- Txted 73, 214
- TYP 99, 153
- Type.Changer (utility) 214-215

- U

- Ultraterm information 165
- Undo (fixing mistakes) 29, 34, 56, 73
- Unknown Label 183
- Unlocking files 19
- Unnew 68, 72
- Upper case / lower case,
 - to change 24, 37
 - labels 88
- USE, 97
 - and Macro Libraries 97
- USER 45, 68, 72
- USR 97, 124-125
- Using Sourceror 184-191
- Utilities 197-215

- V

- VAL 45, 69, 72
- Value of labels 45
- VAR 98
- Variables 79-80, 98, 115, 122, 131, 135
- VER 159
- Vertical Tabs 30, 33, 34, 56, 70, 71
- VIDeo 45, 69

- W

- W: Write Text File, 21
- W (Word - Sourceror) 189
- W 0 command 46, 69
- Word find 25, 28, 30, 39, 40, 50, 56
- Where 45, 69, 72
- Why Macros? 127
- Why a Linker? 137
- Wild Cards,
 - in Delimited Strings 37
 - character, changing the 167
- Word processing text files 44, 67

- X

- XC 84, 123
- XREF 205-207
- XREFA 208

Z

Zero Page Addresses used by Merlin for USR
commands 124

Zero page addressing, forced 85

ASCII Character Chart

Hex	Dec	Binary	Hex	Dec	Binary	ASCII Character
\$00	0	0000 0000	\$80	128	1000 0000	^@ NUL Null
\$01	1	0000 0001	\$81	129	1000 0001	^A SOH Start of heading
\$02	2	0000 0010	\$82	130	1000 0010	^B STX Start of text
\$03	3	0000 0011	\$83	131	1000 0011	^C ETX End of text
\$04	4	0000 0100	\$84	132	1000 0100	^D EOT End of transmission
\$05	5	0000 0101	\$85	133	1000 0101	^E ENQ Enquiry
\$06	6	0000 0110	\$86	134	1000 0110	^F ACK Acknowledge
\$07	7	0000 0111	\$87	135	1000 0111	^G BEL Bell
\$08	8	0000 1000	\$88	136	1000 1000	^H BS Backspace
\$09	9	0000 1001	\$89	137	1000 1001	^I HT Horiz. TAB
\$0A	10	0000 1010	\$8A	138	1000 1010	^J LF Linefeed
\$0B	11	0000 1011	\$8B	139	1000 1011	^K VT Up Arrow
\$0C	12	0000 1100	\$8C	140	1000 1100	^L FF Formfeed
\$0D	13	0000 1101	\$8D	141	1000 1101	^M CR Return
\$0E	14	0000 1110	\$8E	142	1000 1110	^N SO Shift out
\$0F	15	0000 1111	\$8F	143	1000 1111	^O SI Shift in
\$10	16	0001 0000	\$90	144	1001 0000	^P DLE Data link escape
\$11	17	0001 0001	\$91	145	1001 0001	^Q DC1 XON
\$12	18	0001 0010	\$92	146	1001 0010	^R DC2 AUXON
\$13	19	0001 0011	\$93	147	1001 0011	^S DC3 XOFF
\$14	20	0001 0100	\$94	148	1001 0100	^T DC4 AUXOFF
\$15	21	0001 0101	\$95	149	1001 0101	^U NAK Negative Ack. (Right Arrow)
\$16	22	0001 0110	\$96	150	1001 0110	^V SYN Synchronous file
\$17	23	0001 0111	\$97	151	1001 0111	^W ETB End of transmission block
\$18	24	0001 1000	\$98	152	1001 1000	^X CAN Cancel line
\$19	25	0001 1001	\$99	153	1001 1001	^Y EM End of medium
\$1A	26	0001 1010	\$9A	154	1001 1010	^Z SUB Substitute
\$1B	27	0001 1011	\$9B	155	1001 1011	^[ESC Escape
\$1C	28	0001 1100	\$9C	156	1001 1100	^\ FS File or form separator
\$1D	29	0001 1101	\$9D	157	1001 1101] GS Group separator
\$1E	30	0001 1110	\$9E	158	1001 1110	^^ RS Record separator
\$1F	31	0001 1111	\$9F	159	1001 1111	^_ US Unit separator
\$20	32	0010 0000	\$A0	160	1010 0000	space
\$21	33	0010 0001	\$A1	161	1010 0001	!
\$22	34	0010 0010	\$A2	162	1010 0010	"
\$23	35	0010 0011	\$A3	163	1010 0011	#
\$24	36	0010 0100	\$A4	164	1010 0100	\$
\$25	37	0010 0101	\$A5	165	1010 0101	%
\$26	38	0010 0110	\$A6	166	1010 0110	&
\$27	39	0010 0111	\$A7	167	1010 0111	'
\$28	40	0010 1000	\$A8	168	1010 1000	(
\$29	41	0010 1001	\$A9	169	1010 1001)
\$2A	42	0010 1010	\$AA	170	1010 1010	*
\$2B	43	0010 1011	\$AB	171	1010 1011	+
\$2C	44	0010 1100	\$AC	172	1010 1100	,
\$2D	45	0010 1101	\$AD	173	1010 1101	-
\$2E	46	0010 1110	\$AE	174	1010 1110	.
\$2F	47	0010 1111	\$AF	175	1010 1111	/
\$30	48	0011 0000	\$B0	176	1011 0000	0
\$31	49	0011 0001	\$B1	177	1011 0001	1
\$32	50	0011 0010	\$B2	178	1011 0010	2
\$33	51	0011 0011	\$B3	179	1011 0011	3
\$34	52	0011 0100	\$B4	180	1011 0100	4
\$35	53	0011 0101	\$B5	181	1011 0101	5
\$36	54	0011 0110	\$B6	182	1011 0110	6
\$37	55	0011 0111	\$B7	183	1011 0111	7
\$38	56	0011 1000	\$B8	184	1011 1000	8
\$39	57	0011 1001	\$B9	185	1011 1001	9
\$3A	58	0011 1010	\$BA	186	1011 1010	:
\$3B	59	0011 1011	\$BB	187	1011 1011	;
\$3C	60	0011 1100	\$BC	188	1011 1100	<
\$3D	61	0011 1101	\$BD	189	1011 1101	=
\$3E	62	0011 1110	\$BE	190	1011 1110	>
\$3F	63	0011 1111	\$BF	191	1011 1111	?

\$40	64	0100	0000	\$C0	192	1100	0000	e
\$41	65	0100	0001	\$C1	193	1100	0001	A
\$42	66	0100	0010	\$C2	194	1100	0010	B
\$43	67	0100	0011	\$C3	195	1100	0011	C
\$44	68	0100	0100	\$C4	196	1100	0100	D
\$45	69	0100	0101	\$C5	197	1100	0101	E
\$46	70	0100	0110	\$C6	198	1100	0110	F
\$47	71	0100	0111	\$C7	199	1100	0111	G
\$48	72	0100	1000	\$C8	200	1100	1000	H
\$49	73	0100	1001	\$C9	201	1100	1001	I
\$4A	74	0100	1010	\$CA	202	1100	1010	J
\$4B	75	0100	1011	\$CB	203	1100	1011	K
\$4C	76	0100	1100	\$CC	204	1100	1100	L
\$4D	77	0100	1101	\$CD	205	1100	1101	M
\$4E	78	0100	1110	\$CE	206	1100	1110	N
\$4F	79	0100	1111	\$CF	207	1100	1111	O
\$50	80	0101	0000	\$D0	208	1101	0000	P
\$51	81	0101	0001	\$D1	209	1101	0001	Q
\$52	82	0101	0010	\$D2	210	1101	0010	R
\$53	83	0101	0011	\$D3	211	1101	0011	S
\$54	84	0101	0100	\$D4	212	1101	0100	T
\$55	85	0101	0101	\$D5	213	1101	0101	U
\$56	86	0101	0110	\$D6	214	1101	0110	V
\$57	87	0101	0111	\$D7	215	1101	0111	W
\$58	88	0101	1000	\$D8	216	1101	1000	X
\$59	89	0101	1001	\$D9	217	1101	1001	Y
\$5A	90	0101	1010	\$DA	218	1101	1010	Z
\$5B	91	0101	1011	\$DB	219	1101	1011	{
\$5C	92	0101	1100	\$DC	220	1101	1100	\
\$5D	93	0101	1101	\$DD	221	1101	1101	}
\$5E	94	0101	1110	\$DE	222	1101	1110	^
\$5F	95	0101	1111	\$DF	223	1101	1111	~
\$60	96	0110	0000	\$E0	224	1110	0000	
\$61	97	0110	0001	\$E1	225	1110	0001	a
\$62	98	0110	0010	\$E2	226	1110	0010	b
\$63	99	0110	0011	\$E3	227	1110	0011	c
\$64	100	0110	0100	\$E4	228	1110	0100	d
\$65	101	0110	0101	\$E5	229	1110	0101	e
\$66	102	0110	0110	\$E6	230	1110	0110	f
\$67	103	0110	0111	\$E7	231	1110	0111	g
\$68	104	0110	1000	\$E8	232	1110	1000	h
\$69	105	0110	1001	\$E9	233	1110	1001	i
\$6A	106	0110	1010	\$EA	234	1110	1010	j
\$6B	107	0110	1011	\$EB	235	1110	1011	k
\$6C	108	0110	1100	\$EC	236	1110	1100	l
\$6D	109	0110	1101	\$ED	237	1110	1101	m
\$6E	110	0110	1110	\$EE	238	1110	1110	n
\$6F	111	0110	1111	\$EF	239	1110	1111	o
\$70	112	0111	0000	\$F0	240	1111	0000	p
\$71	113	0111	0001	\$F1	241	1111	0001	q
\$72	114	0111	0010	\$F2	242	1111	0010	r
\$73	115	0111	0011	\$F3	243	1111	0011	s
\$74	116	0111	0100	\$F4	244	1111	0100	t
\$75	117	0111	0101	\$F5	245	1111	0101	u
\$76	118	0111	0110	\$F6	246	1111	0110	v
\$77	119	0111	0111	\$F7	247	1111	0111	w
\$78	120	0111	1000	\$F8	248	1111	1000	x
\$79	121	0111	1001	\$F9	249	1111	1001	y
\$7A	122	0111	1010	\$FA	250	1111	1010	z
\$7B	123	0111	1011	\$FB	251	1111	1011	{
\$7C	124	0111	1100	\$FC	252	1111	1100	
\$7D	125	0111	1101	\$FD	253	1111	1101	}
\$7E	126	0111	1110	\$FE	254	1111	1110	~
\$7F	127	0111	1111	\$FF	255	1111	1111	Delete/rubout

