

Merlin 8/16™

The Complete Macro Assembler System
For the Apple® IIgs, 128K IIe and IIc

Merlin Utilities

```
Q-Quit
C-Catalog
X-Copy files
T-Type files
L-Lock files
U-Unlock files
D-Delete files
E-Execute files
U-Verify files
P-Rename files
TAB-Menu change
S-Sort directory
F-Change file date
/ -Create directory
N-Show volume names
B-Toggle bell, now ON
P-Toggle prompting, now ON
```

Thursday 9-Feb-89 12:4

Merlin-16r 4.00

Copyright 1988 by Glen Bredon
Thursday 9-FEB-89 12:36:57 PM

```
C-Catalog
L-Load source
S-Save source
A-Append file
D-Disk command
F-Full screen editor
O-O Save object code
U-Utility menu
Q-Quit
Source: A90040.L0000
```

Prefix: /HARD/MERLIN GS/



TM

Roger Wagner™
PUBLISHING, INC.

Merlin 8/16

**Complete Macro Assembler System
For the Apple IIgs and 128K IIe/IIc**

by Glen Bredon

Manual by Roger Wagner and Tom Burns

Produced by:

Roger Wagner Publishing, Inc.

1050 Pioneer Way, Suite P
El Cajon, California 92020

Customer Service & Technical Support:
619/442-0522

ISBN 0-927796-28-7
1M188

Customer Licensing Agreement

The Roger Wagner Publishing, Inc. software product that you have just received from Roger Wagner Publishing, Inc., or one of its authorized dealers, is provided to you subject to the Terms and Conditions of the Software Customer Licensing Agreement. Should you decide that you cannot accept these Terms and Conditions, then you must return your product with all documentation and this License marked "REFUSED" within the 30 day examination period following the receipt of the product.

1. License. Roger Wagner Publishing, Inc. hereby grants you upon your receipt of this product, a nonexclusive license to use the enclosed Roger Wagner Publishing, Inc. product subject to the terms and restrictions set forth in this License Agreement.
2. Copyright. This software product, and its documentation, is copyrighted by Roger Wagner Publishing, Inc. You may not copy or otherwise reproduce the product or any part of it except as expressly permitted in this License.
3. Restrictions on Use and Transfer. The original and any backup copies of this product are intended for your personal use in connection with a single computer. You may not distribute copies of, or any part of, this product without the express written permission of Roger Wagner Publishing, Inc.

Limitations of Warranties and Liability

Roger Wagner Publishing, Inc. and the program author shall have no liability or responsibility to purchaser or any other person or entity with respect to liability, loss or damage caused or alleged to be caused directly or indirectly by this software, including, but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of this software. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

ProDOS and DOS 3.3 are copyrighted programs of Apple Computer, Inc. licensed to Roger Wagner Publishing, Inc. to distribute for use only in combination with Merlin 8/16.

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Copyrights

The Merlin 8/16 documentation and software are copyrighted © 1987 by Roger Wagner Publishing, Inc. This documentation and/or software, or any part thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, without the prior written permission of the publisher.

We have tried our best to give you a quality product at a fair price, and made the software copyable for your personal convenience. Please recommend our product to your friends, but respect our wishes to not make copies for others. Thanks!

Apple, ProDOS, and DOS 3.3 are trademarks of Apple Computer, Inc.

Merlin 8/16 is a trademark of Roger Wagner Publishing, Inc.

About the Manual

This manual was formatted using MacAuthor from Icon Technology, Ltd, 9 Jarrom Street, Leicester LE2 7DH, England, and Apple's LaserWriter printer.

OUR GUARANTEE

This product carries the unconditional guarantee of satisfaction or your money back. Any product may be returned to place of purchase for complete refund or replacement within thirty (30) days of purchase if accompanied by the sales receipt or other proof of purchase.

ABOUT THE AUTHOR

Glen Bredon is a professor at Rutgers University in New Jersey where he has taught mathematics for over twenty years. He purchased his first computer in 1979 and began exploring its internal operations because "I wanted to know more than my students." The result of this study was Merlin, the first in a series of best selling assemblers (Merlin, Merlin Pro, Merlin 64, and Merlin 128) for the Apple and Commodore personal computers. Glen has also written other utilities including Prosel, the popular ProDOS program selector. A native Californian and concerned environmentalist, Glen spends his summers away from mathematics and computing, preferring the solitude of the Sierra Nevada mountains where he has helped establish wilderness reserves.



MERLIN 8/16

Table of Contents

INTRODUCTION

System Requirements	1
Suggested Reading	2
Making Back-up Copies of Merlin 8/16	3
Personalizing Merlin 8/16	4

GETTING STARTED WITH MERLIN 8/16

Entering a Source Listing	5
Editing a Source Listing	8
Assembling a Source Listing	9
Saving Programs	12
Running Programs	14
The Rest of This Manual	16

THE MAIN MENU

The Main Menu	17
-------------------------	----

THE MERLIN 8 EDITOR

Merlin 8 Full Screen Editor Commands	23
Merlin 8 Control Key Commands	24
Merlin 8 Open Apple Key Commands	27
Merlin 8 Editor Command Summary	33
Merlin 8 Command Mode	36
General Guidelines for the Command Mode	36
About the Merlin 8 Editor Documentation	36
Command Mode Commands	37
Assembling a Merlin 8 File	47

THE MERLIN 16 EDITOR

Merlin 16 Full Screen Editor Commands	48
Merlin 16 Control Key Commands	48
Merlin 16 Open Apple Key Commands	52
Merlin 16 Command Box	61
General Guidelines for the Command Box Commands	61
About the Command Box Documentation	61
Command Box Commands	63
Merlin 16 Editor Command Summary	70
Assembling a Merlin 16 File	73

THE ASSEMBLER

The Assembler	74
About the Assembler Documentation	75
Preliminary Definitions	76
Assembler Syntax Conventions	77
Source Code Format	77
Local Labels	78
Variables	79
Number Format	80
Expressions Allowed by the Assembler	81
Immediate Data Syntax	83
Addressing Modes	83
Special Forced Non-Zero Page Addressing	85
Sweet 16 Opcodes	85
65C02 and 65802 Opcodes	86
Assembler Pseudo Opcode Descriptions	87
Formatting Pseudo Ops	103
String Data Pseudo Ops	107
Data and Storage Allocation Pseudo Ops	110
Using Data Tables in Programs	113
Conditional Pseudo Ops	114
Using Conditional Assembly	116
Miscellaneous Pseudo Ops	119

MACROS

Why Macros	127
Macro Pseudo Ops	127
How a Macro Works	128
More About Defining a Macro	133
Nested Macros	134
Macro Libraries and the USE Pseudo-Op	136

THE LINKERS

The Merlin 8/16 Linkers	137
Why a Linker	137
About the Linker Documentation	138
Pseudo Opcodes for Use with Relocatable Code Files	140
The Merlin 8 Linker	144
Linker Name Files	145
The Linking Process	146
Using Merlin 8 Linked Files	147
The Merlin 16 Linkers	150
The Absolute Linker	151
The Linker Command File	154
Using Merlin 16 Linked Files	155

Multiple Output Files	157
The GS Linker	158
Quick Link	159
Multiple LNK Input Files	160
Using the GS Linker	161
Large File GS Linker	163

TECHNICAL INFORMATION

Technical Information	164
General Information (DOS 3.3 only)	164
General Information (ProDOS and DOS 3.3)	165
Symbol Table	165
Ultraterm Information	165
Memory Allocation with Merlin 8/16	166
Configuring Merlin 8 ProDOS	166
Configuring Merlin 8 DOS 3.3	166
Data Descriptions for Merlin 8 Configurations	166
64K Merlin and Merlin 8/16 Source Files	167
Merlin 8/16 ProDOS Notes	168
Transferring Source Files from DOS 3.3 to ProDOS Merlin 8/16	169
Merlin 8/16 and Speed Up Cards	169
Merlin 8 and Corvus Hard Disks	170
Configuring Merlin 16	170
The Merlin 16 Params File	170
Merlin 8/16 Memory Maps	176

ERROR MESSAGES

Error Messages	179
--------------------------	-----

SOURCEROR

Using Sourceror on Merlin 16	184
Using Sourceror on Merlin 8	184
65C02 Opcodes on Older IIe/IIc Computers	186
Disassembly Commands	186
Merlin 16 Sourceror Notes	188
Sourceror XL	188
Final Processing	190
Modifying the Finished Source	190
The Memory Full Message	191
Changing Sourceror's Label Tables	191

SOURCEROR.FP

Applesoft Source Listing	192
Sourceror.FP	192
Printing the Applesoft Source Listing	193

Applesoft Source Cross Reference Listing	195
--	-----

UTILITY PROGRAMS

Auto Edit	197
Classic Desk Accessories	197
Calendar - Notepad - Rational Calculator	197
Clock	197
Conv Lnk Rel	198
Converter: APW source to Merlin	198
Using Macgen	200
Cross Reference Programs	205
Xref/XrefA	205
Using Xref	206
XrefA Notes	208
Formatter	208
Keyboard Macro Files	209
Edmac - Keymac	209
Keyboard Macro Equivalent Chart	210
Make Dump	211
Printfiler	211
Printfiler Applications	212
Using Printfiler	212
Changing Printfiler Options	213
Txted	214
Type.Changer	214
Additional Merlin 8/16 Resource Files	215

INDEX

Index	216
-----------------	-----

MERLIN 8/16

Merlin 8/16 is an extremely powerful, comprehensive Macro Assembler system for the Apple IIgs or 128K IIe/IIc. It consists of four main modules and numerous auxiliary and utility programs which comprise one of the most complete assembler systems available for *any* personal computer. Merlin 8/16's four main modules are:

- FILE MANAGEMENT system, for disk I/O, file management, ProDOS interpreter, etc.
- EDITOR system, for writing and editing programs with word-processor-like power.
- ASSEMBLER system, with Macros, Macro libraries, conditional assembly, linked files, etc.
- LINKER system, for generating relocatable code modules, library routines, run-time packages, etc.

However, Merlin 8/16 is more than just the sum of these four parts. Here are some of the other features offered by Merlin 8/16:

- Merlin 8/16 comes with *three* assemblers: Merlin 16 (ProDOS), Merlin 8 (ProDOS), and Merlin 8 (DOS 3.3). On the Apple IIgs, or Apple IIe or IIc computers with the 65802 or 65816 chip, Merlin 16 assembles programs written for the 6502, 65C02, 65802 and 65816 microprocessors. On the standard 128K IIe or IIc, Merlin 8 assembles programs written for the 6502, 65C02 and 65802 microprocessors.
- Merlin 8/16 is compatible with existing Merlin and Merlin Pro source files.
- Merlin 8/16 recognizes over 50 Pseudo Opcodes for extreme programming flexibility.
- Merlin 8/16 has over 70 commands for ultimate editing and assembling power.
- Merlin 8/16 produces a fully commented, disassembled source listing of Applesoft BASIC.
- Merlin 8/16 comes with Sourceror, a powerful symbolic disassembler to generate Merlin 8/16 source code from binary programs.
- Merlin 8/16 comes with many sample programs, libraries and other aids to get you going with assembly language fast.
- Merlin 8/16 is copyable and hard disk compatible.

O
D
E
D
O
C
E
R
E
E
E
E
E
E
E
E
E
E
E

INTRODUCTION

Merlin 8/16 is the most comprehensive macro assembler system for the Apple IIgs or 128K IIe/IIc offering virtually every feature and function that a programmer needs, thus making it unlikely that you'll outgrow it. At the same time, Merlin 8/16's easy built-in editor and fast assemblies make it a pleasure to use whether you're writing a few lines of code or 30,000!

SYSTEM REQUIREMENTS

To run Merlin 8/16, you'll need at least one disk drive, and one of the following:

- * Apple IIgs or
- * Apple IIc or
- * 128K Apple IIe or
- * Laser 128 or Laser 128 EX

Merlin 8 will run on all the computer systems listed above; Merlin 16 requires a 65802 or 65816 microprocessor such as in the Apple IIgs, or a modified Apple IIe, IIc or compatible.

Merlin 8/16 supports a wide variety of 80 column display devices including the Apple, the Videx standard and Ultraterm cards, the Checkmate Multi-View, the Applied Engineering Viewmaster 80, and many others.

Merlin 8/16 works with all printers, producing formatted listings with page breaks and titles.

If you're familiar with assembly language programming already, you will find it's easy to adapt to Merlin 8/16. It follows the programming standards of the 65xx family of microprocessors, and its assembler-directed commands, or pseudo-ops, are a super-set of just about every other assembler. That is, assembler directives like HEX, ASC, DS, etc. that you've used in other assemblers are used in Merlin 8/16, and better still, you'll find a new complement of functions to make programming easier. These include assembling directly to or from disk files, multiple data formats for numbers and strings, a complete set of assembler utilities such as cross-referencing and a source code generator (Sourceror), macro capabilities and more.

If you're new to assembly language programming, Merlin 8/16 is the easiest assembler there is. However, the Merlin 8/16 manual does not make any attempt to teach the techniques of assembly language programming itself. Those techniques are covered in various tutorial books available from a number of publishers, including Roger Wagner Publishing. Because everyone has different goals and objectives, you should seek out those books which best match your current needs and experience.

SUGGESTED READING

Some of the books we recommend include:

- ✓ APPLE IIGS MACHINE LANGUAGE FOR BEGINNERS - Roger Wagner, COMPUTE! Books, Greensboro, NC 27408. Approximately 600 pages of new material, this IIGs assembly language tutorial covers a myriad of subjects such as writing your first routines, calling machine language routines from Applesoft, ProDOS 8 and 16, and concludes with a IIGs drawing program using Quickdraw, scrollable windows, and pull-down menus. The encyclopedic appendix includes examples of all machine language instructions.
- ✓ APPLE IIGS TECHNICAL REFERENCE - Michael Fischer, Osbourne-McGraw-Hill. Filled with relevant technical material, this book is an outstanding resource for the advanced programmer interested software development on the IIGs.

APPLE IIGS TOOLBOX REFERENCE - Apple Computer, Inc. This 2-volume set is a highly technical reference for using the IIGs toolbox. Does not include programming examples or tutorial information.
- ✓ APPLE PRODOS - by Gary Little. Brady Communications Co., Bowie, MD 20715. A good tutorial book on how to write assembly language programs that use ProDOS.

ASSEMBLY LINES: THE BOOK - by Roger Wagner. Roger Wagner Publishing, Inc. El Cajon, CA 92020. A tutorial on assembly language programming designed specifically for the novice. It gives a description of all 6502 instructions, disk access, reading and writing DOS 3.3 files, sound generation, basic math, keyboard and screen techniques and more.

ASSEMBLY COOKBOOK FOR THE APPLE II/IIe - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268. This interesting book will give you real insights into Don Lancaster's view of programming theory and practice. A good addition to any library.
- ✓ BENEATH APPLE DOS - by Don Worth and Peter Lechner. Brady Communications Co., Bowie, MD 20715.
- ✓ BENEATH APPLE PRODOS - by Don Worth and Peter Lechner. Brady Communications Co., Bowie, MD 20715. These two books are the classic reference books for learning about the inner workings of DOS 3.3 and ProDOS. A must if you intend to do any programming in this area. More of a reference than a tutorial.

ENHANCING YOUR APPLE II (Volume 1) - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268.

ENHANCING YOUR APPLE II AND IIe (Volume 2) - by Don Lancaster. Howard Sams & Co., Indianapolis, IN 46268. These two books are a continuation of Don Lancaster's unique instruction in the art of assembly language programming. More tips here on different kinds of short programs.

INSIDE THE APPLE IIC - by Gary Little. Brady Communications Co., Bowie, MD 20715. An overview of the entire Apple Iic system, from the assembly language programmer's viewpoint.

INSIDE THE APPLE IIGS - Gary Bond, Sybex. Contains a lot of important information but by-passes some of the rules for producing code compatible with future system upgrades.

NOW THAT YOU KNOW APPLE ASSEMBLY LANGUAGE, WHAT CAN YOU DO WITH IT? - by Jules Guildler, Redlig Systems, Inc., 2068 79th Street, Brooklyn, NY 11214. A good applications tutorial with lots of good subroutines for input and output, printer drivers and more. An excellent follow-up book to Assembly Lines: The Book.

PROGRAMMING THE 65816 - David Eyes and Ron Lichty, Prentice Hall Press, New York, NY 10023. In-depth coverage of the differences between the 65816, 6502, 65C02, and 65802 chips and how to best utilize them from a programming standpoint. Also includes a programming tutorial, code samples and a reference section.

✓ THE APPLE IIGS TOOLBOX REVEALED - Danny Goodman, Bantam Computer Books. Details the philosophy of the toolbox and methods of accessing it, but lacks programming examples.

THE ELEMENTARY APPLE IIGS - William B. Sanders, COMPUTE! Books, Greensboro, NC 27408. An introductory book to the IIGs with information on Applesoft, Hi-Res and Super Hi-Res graphics, and sound on the IIGs.

65816/65802 ASSEMBLY LANGUAGE PROGRAMMING - by Michael Fischer. Osborne McGraw-Hill, Berkely, CA 94710. A thorough treatment of the 6502, 65C02, 65802 and 65816 microprocessors. The size of this book (nearly 700 pages) gives you an idea of why this manual for Merlin 8/16 does not attempt to cover programming on the 6500 family microprocessors itself.

MAKING BACK-UP COPIES OF MERLIN 8/16

The Merlin 8/16 diskettes are unprotected and copies may be made using any copy utility. It is highly recommended that you use *only* the BACK-UP copy of Merlin 8/16 in your daily work, and keep the original in a safe place.

You can copy the Merlin 8/16 diskettes using:

- 1) The COPYA utility program from Apple's System Master Diskette for the DOS 3.3 version of Merlin 8/16.
- 2) The Copy Volume or Duplicate a Diskette function from the ProDOS System Disk for the ProDOS versions of Merlin 8/16, or System Utilities program on the Apple IIGS System Master diskette.

PERSONALIZING MERLIN 8/16

Certain aspects of Merlin 8/16, such as printer line width, page length, default tab positions for the fields in a source listing, turning off the bell sound, etc., can be customized by changing the file **PARMS.S** on the Merlin 8/16 disk, and re-assembling the **PARMS** file. If you would like to change any of these defaults of Merlin 8/16, see the discussion of the **PARMS** file in the Technical Information section of this manual for details on making the changes. For now, though, we recommend you wait until you're more familiar with Merlin 8/16.

GETTING STARTED WITH MERLIN 8/16

The purpose of this section is not to provide instruction in assembly language programming. Rather, it will show you the entry and running of a short assembly language program to give you an idea of how Merlin 8/16 works.

Many of the Merlin 8/16 commands and functions are very similar in operation. This section does not attempt to present demonstrations of each and every command option. The objective is to present examples of the more common operations, sufficient to get you started writing your own programs using Merlin 8/16. You should not expect to immediately use all of the various commands that Merlin 8/16 supports in your first program. The best approach is to use the Merlin 8/16 manual in an encyclopedia-like fashion, reading just those sections that provide some utility to a current programming task. We suggest that you lightly skim through the manual once, to become aware of generally what the software has to offer, and then return later to specific sections as needed.

ENTERING A SOURCE LISTING

Now, let's try your first program with Merlin 8/16. Just follow these steps:

1. Start any of the Merlin 8/16 disks (ProDOS or DOS 3.3). A title screen appears, after which the screen changes to the Main Menu. The Main Menu is used for loading and saving files, disk operations, and of course, entering the Merlin 8/16 Editor and Assembler itself.
2. The percent (%) prompt appears at the bottom of the Main Menu. Press F if you are using Merlin 16. If you are using Merlin 8, press E to go the Editor, then A and Return to enter the Add Mode.
3. Since we are entering an entirely new program, a 1 appears at the top right corner of the screen. This number indicates the line the cursor is on in the listing. The 1 and all subsequent line numbers which appear serve roughly the same purpose as line numbers in BASIC except that in assembly source code, line numbers are not referenced for jumps to subroutines or in GOTO-like statements.
4. On line 1, type an asterisk (*). Entering an asterisk as the first character in any line is similar to a REM statement in BASIC - it tells the assembler this is a remark line and anything after the asterisk is to be ignored. Type the title DEMO PROGRAM 1 after the asterisk and press the Return key.

```
* DEMO PROGRAM 1
```

5. After Return, the cursor moves to the beginning of line 2.
6. An asterisk is not needed to just create a blank line. To create a blank line, with no text following it, press Return again.
7. The cursor once again drops down one line, and a 3 appears at the top right corner of the screen.

8. Press the space bar once and the cursor moves to the next field. Type `ORG`, press space again, type `$8000` and then press Return.

```
* DEMO PROGRAM 1
```

```
    ORG    $8000
```

The above step instructs the assembler to create the following program so that it can run at memory location `$8000`. Merlin 8/16 almost always assembles your program in the same place in memory, but the `ORG` (for Origin) is used to tell Merlin 8/16 where you want the program to eventually be run. This is so that `JMPs`, `JSRs` and other location dependent code within your program is properly written with the final location in mind.

You'll notice that when you press the space bar, Merlin 8/16 automatically moves the cursor to the next field on the line. You'll recall that in assembly language programming, the position of text on each line determines what kind of information it is. Labels for routines and entry points are in the first position. On line 3 you skipped this field by pressing the space bar first, before entering any text. The second position is for the command itself. The command can either be a command such as `LDA`, `RTS`, etc., or it can be a directive to Merlin 8/16 itself, to be used during the assembly to write a file to disk, create a label, call up a macro, or any of Merlin 8/16's many assembler commands.

9. With the cursor at the beginning of line 4, type `BELL` and space to the next field, type `EQU` and space again, then type `$FBDD` and press Return.

```
    BELL  EQU  $FBDD
```

This defines the label `BELL` to be equal to hex `FBDD`. This use of a label is known as an equate or constant. Wherever `BELL` appears in an expression, it will be replaced with `$FBDD`. Why don't we just use `$FBDD`? For one thing, `BELL` is easier to remember than `$FBDD`. Also, if a later assembly required changing the location of `BELL`, all that needs changing is the `EQU` statement, rather than all the other `$FBDD`'s throughout the listing.

10. At the beginning of line 5, type `START` and press space, type `JSR` and space again, type `BELL` and another space, then type a semicolon (;) followed by `RING THE BELL` and then press Return.

```
    START JSR  BELL      ; RING THE BELL
```

Following the opcode is the operand, in this case `BELL`. The operand is the target information of the opcode. Where to `JSR` to, what value to load, etc.

Semicolons are like asterisks, used to mark a comment. Semicolons, however, are used to mark the start of a comment at the end of a line that contains other commands.

11. On line 6 type `DONE` and space, then type `RTS` and press Return.

12. The program has been completely entered. If you wanted to exit the Add Mode, you would press Open-Apple-Q ("Quit"). Since you are not done, *do not exit yet*.

13. The screen should now appear like this:

```
* DEMO PROGRAM 1

      ORG   $8000
BELL   EQU   $FBDD
START  JSR   BELL       ; RING THE BELL
DONE   RTS
```

Note that throughout the entry of this program, each bit of text has been moved to a specific field. Here is a summary of the fields as used so far:

Label	Opcode	Operand	; Comment
START	JSR	BELL	; RING THE BELL

Field One is reserved for labels. START is an example of a label.

Field Two is reserved for opcodes, such as the Merlin 8/16 pseudo-opcodes or directives such as ORG and EQU, and the JSR and RTS opcodes.

Field Three is for operands, such as \$8000, \$FBDD and, in this case, BELL.

Field Four contains comments which are preceded by a semi-colon (;).

It should be apparent from this exercise that it is not necessary to input extra spaces in the source file for formatting purposes, even if these spaces seem to exist in a listing you may be using.

In summary, on each line:

- 1) Do not space for a label. Space once after a label or if there is no label, once at the beginning for the opcode.
- 2) Space once after the opcode for the operand. Space once after the operand for the comment. If there is no operand, type a space and a semicolon for a comment if desired.

EDITING A SOURCE LISTING

Assuming no errors have been made in the text entered so far, you could now assemble the program entered with Merlin 8/16. Before doing that, however, let's look at the editing abilities of Merlin 8/16.

Editing is the process of making alterations to text that you've already entered, and this ability is one of Merlin 8/16's strong points. In a sense, an assembler is just a word processor for the text that makes up a program. In that light, then, you can judge an assembler in part by how good its editing features are.

Merlin 8/16 has a powerful Full Screen Editor. Powerful in the range of operations possible and, after a little practice, remarkably easy to use. The following text describes how to use the Editor.

You can use the arrow keys to move the cursor anywhere in the listing. There are two types of cursors, the insert and the overstrike. How and when to switch between these two cursor will be explained later. For now, we'll use the insert cursor.

Inserting and deleting lines in the source code are both simple operations. The following example will insert three new lines between the existing lines 5 and 6.

1. Use the arrow key to move to the beginning of line 4 (BELL etc.). Press Return to insert a blank line.
2. Press Return.
3. Press Return again.
4. At line 7, press space once, then type TYA and press Return.
5. The listing should appear as follows:

```
* DEMO PROGRAM 1

      ORG   $8000
BELL  EQU   $FBDD

      TYA
START JSR   BELL      ; RING THE BELL
DONE  RTS
```

The three new lines (5, 6, and 7) have been inserted, and the subsequent original source lines (now lines 8 and 9) have been moved down.

Deleting lines is equally easy.

1. In Merlin 16, press a key to re-enter the Full Screen Editor.

In Merlin 8, press A and Return to enter the Add Mode (assuming you are at the command mode).

2. Move the cursor to the beginning of line 8 (START etc.) and press **Open-Apple-Delete**.

You've just deleted the TYA line, and the subsequent lines have been renumbered.

Open-Apple-Delete always deletes the line *above* the cursor. (**Open-Apple-D** will delete the line the cursor is on.)

3. Press **Open-Apple-Delete** twice more.

Lines 5 and 6 from the previous example have been deleted, and the subsequent lines renumbered. The listing appears the same as when you first entered it into to the Editor.

While adding, editing, or deleting an existing line, you have many options within the line, all of which are accessed by using Control characters. To demonstrate using our BELL routine:

1. Move the cursor to the beginning of line 6. The cursor should be over the D in DONE.
2. Type **Control-D**. The character under the cursor disappears, and the text of the label moves to the left. Type **Control-D** again, and yet a third and fourth time. DONE has been deleted, and the cursor is positioned at the beginning of the label field.
3. Type DONE in again and note that the text is inserted in the label field but the opcode field does not move. The listing appears the same as when you first entered it into the Editor.

The other Control Character commands function similarly. All of the editing commands are discussed in the Merlin 16 and Merlin 8 Editor sections of this manual.

ASSEMBLING A SOURCE LISTING

The next step in using Merlin 8/16 is to assemble the source code into object code.

1. If you are using Merlin 16, press **Open-Apple-A** to assemble and skip to step 3.

If you are using Merlin 8, press **Open-Apple-Q** to enter the Command Mode. After the colon (:)
prompt, type **ASM** and press Return.

2a. The Merlin 16 screen appears as follows:

Assembling.

```

          1      * DEMO PROGRAM 1
          2
          3          ORG  $8000
          4      BELL    EQU  $FBDD
008000 20 DD FB      5      START    JSR  BELL    ; RING THE BELL
008003 60          6      DONE     RTS

```

End Merlin-16 assembly, 4 bytes, errors: 0 , symbol table: \$1800-\$181D

Symbol table - alphabetical order:

```

    BELL    =FBDD    ? DONE    =8003    ? START    =8000

```

Symbol table - numerical order:

```

? START    =8000    ? DONE    =8003    BELL    =FBDD

```

Press a key.

2b. The Merlin 8 screen appears as follows:

Assembling

```

          1      * DEMO PROGRAM 1
          2
          3          ORG  $8000
          4      BELL    EQU  $FBDD
8000 20 DD FB      5      START    JSR  BELL    ; RING THE BELL
8003 60          6      DONE     RTS

```

--End assembly, 4 bytes, Errors: 0

Symbol table - alphabetical order:

```

    BELL    =FBDD    ? DONE    =8003    ? START    =8000

```

Symbol table - numerical order:

```

? START    =8000    ? DONE    =8003    BELL    =FBDD

```

:

If instead of completing the above listing, the system beeps and displays an error message, note the line number referenced in the message, and press Return until the "End assembly..." message appears. Then refer back to the section where the program was first entered and compare the listing with the one shown in this manual. Look especially for elements in incorrect fields. Using the editing functions you've learned, change any lines in your listing which do not look like those in the listing, then assemble again.

If all went well, to the right of the column of line numbers down the middle of the screen is the now familiar, formatted *source code*.

To the left of the line numbers is a series of numeric and alphabetic characters. This is the *object code*, which are the opcodes and operands assembled to their machine language hexadecimal equivalents.

Merlin 16 example:

```
008000 20 DD FB      5   START   JSR  BELL   ;ring the bell
```

Merlin 8 example:

```
 8000 20 DD FB      5   START   JSR  BELL   ;ring the bell
```

Left to right, starting on line 5, the first group of characters is the routine's starting address in memory. See the definition of `ORG` in the section on the Assembler. On the left of line 5, the number 20 appears after the colon. This is the one-byte hexadecimal code for the opcode `JSR`.

NOTE: The label `START` is not assembled into object code; neither are comments, remarks, or pseudo-ops such as `ORG`. Such elements are for the convenience and utility of the programmer only and the use of the assembler program.

Each pair of hexadecimal digits is one byte. The next two bytes on line 5 bear a curious resemblance to the last group of characters on line 4; have a look. In line 4 of the source code we told the assembler that the label `BELL` was `EQU`ated to address `$FBDD`. In line 5, when the assembler encountered `BELL` as the operand, it substituted the specified address. The sequence of the high- and low-order bytes was reversed, turning `$FBDD` into `DD FB`. This is a 65xx microprocessor convention.

The rest of the information presented should explain itself. The total errors encountered in the source code was zero. If you count the bytes following the addresses, you'll see there were four bytes of object code generated.

SAVING PROGRAMS

There is an important step before running a program you have assembled. You should always save the source code first in case your program crashes when you run it and causes your computer to hang. If you save the program first, you'll be able to load it again and continue editing it. To save the source code, you will have to return to the Main Menu, and use the SAVE SOURCE command. Then you would use the SAVE OBJECT CODE. Note that SAVE OBJECT CODE can only be used if there has been a successful assembly.

Here are the steps to follow for Merlin 16:

1. After a successful assembly, Merlin 16 returns automatically to the Main Menu. (You can also return to the Main Menu in Merlin 16 by pressing **Open-Apple-Q**) If the Merlin 16 system disk is still in the drive, remove it and insert an initialized data disk.

Press D for DISK COMMAND, then type PREFIX followed by the pathname for your data disk. For example, if the volume name of your data disk was MYDISK, the complete line would look like this:

Disk command:PREFIX/MYDISK

When you press Return, Merlin 16 will change the current prefix in the Main Menu box.

2. After the per cent (%) prompt, press S to SAVE SOURCE. The system is now waiting for a filename. Type DEMO1 and press Return. After the program has been saved, the prompt returns.
3. Press C and Return to catalog the data diskette. The source code has been saved as DEMO1.S and is a text file. The .S suffix is automatically appended by Merlin 16 to the filename by the SAVE SOURCE command. This is a file-labeling convention which indicates the subject file is source code.
4. After the per cent (%) prompt, press Return to go to the Main Menu and press O for SAVE OBJECT CODE. To avoid confusion, the object file should be saved under the same name as was earlier specified for the source file. Press Y to accept DEMO1 as the object name. The object code file is saved as a BIN file.

NOTE: There is no danger of overwriting the source file because no suffix is appended to the object code file name. In our example, the object file will be saved as DEMO1.

5. Press C to catalog again and note the files titled DEMO1.S and DEMO1.

Here are the steps to follow for **Merlin 8**:

1. From the Command Mode prompt (:), press Q to Quit and Return. The system will quit the Editor and go to Main Menu. If the Merlin 8 system disk is still in the drive, remove it and insert an initialized data disk. If you are using the DOS 3.3 version of Merlin 8, skip to step 2.

On the ProDOS version of Merlin 8, press D for DISK COMMAND, then type PREFIX followed by the pathname for the data disk. For example, if the volume name of your data disk was MYDISK, the complete line would look like this:

Disk command:PREFIX/MYDISK

When you press Return, Merlin 8 will change the current prefix on the screen.

2. After the per cent (%) prompt, press S to SAVE SOURCE. The system is now waiting for a filename. Type DEMO1 and press Return. After the program has been saved, the prompt returns.
3. Press C to catalog the diskette. The source code has been saved as DEMO1.S and is a text file on the ProDOS version, or a binary file on the DOS 3.3 version of Merlin 8. The .S suffix is automatically appended by Merlin 8 to the filename by the SAVE SOURCE command. This is a file-labeling convention which indicates the subject file is source code.
4. Press Return to go to the Main Menu and type O for SAVE OBJECT CODE. To avoid confusion, the object file should be saved under the same name as was earlier specified for the source file. Press Y to accept DEMO1 as the object name.

NOTE: There is no danger of overwriting the source file because no suffix is appended to the object code file name. In our example, the object file will be saved as DEMO1.

5. Press C to catalog again and note the files titled DEMO1.S and DEMO1.

When looking at the Main Menu, you'll also notice that the address and length of your source code text is displayed. If you have done a successful assembly, the address and length of the assembled object code is also displayed. If the object code information is not displayed, Merlin 8/16 will not let you save an object file to disk. The object code save is disabled whenever an error has occurred during an assembly, or you have made a change to the source code and not yet re-assembled it, or the source code is either too big to fit or not allowed to reside in the memory range you have specified for the assembly.

See the sections on ORG, OBJ and Memory allocation if this latter problem occurs.

RUNNING PROGRAMS

There are two methods of running programs after you have completed a successful assembly. *Method A is the recommended procedure.* It requires a little more time but it is much safer.

Method A: Running a program from BASIC

1. Save the source code.
2. Save the object code.
3. Press Q to Quit.
4. Load BASIC if necessary.
5. BLOAD the object file from the Applesoft BASIC prompt (>).).

If you are using ProDOS, be sure to include the complete pathname. In our example, this would be:

BLOAD/MYDISK/DEMO1 (Return)

If you are using DOS 3.3, you would type:

BLOAD DEMO1 (Return)

6. To run the program, type CALL followed by the appropriate load address. The DEMO1 program had an ORG of \$8000 which is 32768 in decimal. Thus, from the Applesoft prompt you would type:

CALL 32768 (Return)

If you are running DEMO1, you will hear the beep indicating DEMO1 actually works.

Method B: Running a program from the Monitor

1. If you are using Merlin 16, return to the Editor if necessary by pressing F.

If you are using Merlin 8, return to the Editor if necessary by pressing E.

2. From the Merlin 16 Editor, press Open-Apple-O to open the Command Box.

From the Merlin 8 Editor, press Open-Apple-Q to enter the Command Mode (:).

3. Type GET \$8000 and press Return where \$8000 is the address used in the ORG statement. The GET command tells Merlin 8/16 to take the program you've just assembled and transfer it to Main Memory at the specified location.
4. From the Merlin 16 Editor, press Open-Apple-O to open the Command Box, and type MON and press Return and the Monitor prompt (*) appears.

In Merlin 8, type MON and press Return and the Monitor prompt (*) appears.

5. Type the ORG address followed by a G and press Return to run the program. In our example, you would type 8000G and press Return. A beep is heard. The demonstration program DEMO1 was responsible for it. It works!

NOTE: With Method B, do not forget to use the GET command to move the code to Main Memory. If you assemble your program and then go directly to the Monitor *you will not see your program*. It only gets there after you move it to Main Memory or run it. Also, if your program is loaded in one location but runs in another, you must run the program before you can use the Monitor to examine the code at the final location.

6. You can return to the Main Menu from the Monitor by pressing Control-Y and Return.

APPLE IIGS USERS: Method B *cannot* be used to test programs that call Applesoft or Monitor routines in the address range \$D000-\$FFFF (such as DEMO PROGRAM #1 that calls the BELL routine at \$FBDD). This is because on the GS, these routines are actually located in bank \$FF of memory, while the routine you're testing, at that point, is in bank \$00. When your routine calls the Applesoft or Monitor routine, it will not jump to the proper bank, and the results are unpredictable. You can use Method A to test this type of program, or use whatever instructions may be provided for testing the program, as will likely be the case with listings from magazines or other sources.

THE REST OF THIS MANUAL...

The preceding section was a simple look at how to enter, assemble, save, and run a Merlin 8/16 program.

The remainder of this manual is an encyclopedic reference of the various commands that are available within Merlin 8/16 to make writing an assembly language program easier. Remember that the commands and directives available within Merlin 8/16 are merely the building blocks from which you can create your own programs. It is up to you to decide when and where they are to be used.

The manual describes the following aspects of Merlin 8/16:

- 1) **The Main Menu:** This level of Merlin 8/16 is used for loading and saving files, disk operations, and entering the Editor/Assembler.
- 2) **The Merlin 16 Editor and The Merlin 8 Editor:** These sections describes the functions available for creating and editing a source listing for an assembly language program.
- 3) **The Assembler:** This section covers assembler directives within Merlin 8/16. Remember that these are not editing or direct user commands, but rather, text commands included within a source listing to tell the assembler to do something special while your program is being assembled. This might include using a Macro definition, writing a file to disk, or other functions.
- 4) **Supplemental Sections:** There are a number of additional sections in this manual that describe the use of Macros, the Relocating Linker, Error Messages, Sourceror and Utility Programs, and many other aspects of Merlin 8/16's operation. These can be consulted as necessary.

THE MAIN MENU

The Merlin 8/16 Main Menu is used for file maintenance operations such as loading or saving code or cataloging the disk. The following sections summarize each command available in this mode.

C (Catalog - ProDOS)

When you press C, you will be asked for the pathname of the directory you wish to catalog. At the Prefix: prompt, enter a pathname or press Return. The catalog of the current directory will be shown. The Main Menu prompt (%) is displayed after the catalog is shown. You can then issue any Main Menu command such as L for Load Source. This permits you to give a Main Menu command while the catalog is still on the screen. In addition, if *any* key is typed during the catalog printing, the ProDOS catalog will pause until any other key is pressed.

After using the C command to show the catalog, you can press =, =1, or =2 where the number corresponds to the desired drive number. Merlin 8/16 will set the prefix to the volume found in the current or specified drive and then catalog that volume.

If you press Open-Apple during the catalog, Merlin 8/16 lists only the directory files present in the specified pathname.

If you press Closed-Apple during the catalog, Merlin 8/16 lists only the Text (usually source) files present.

If you press *both* Open- and Closed-Apple keys simultaneously during a catalog, Merlin 8/16 lists only the BIN (usually object) files present. Note that these keys must be pressed and held throughout the entire catalog listing process.

If you enter a 1 as the first character of a pathname, or just 1 and Return, then the catalog will be sent to the printer in slot 1.

If Merlin 8/16 cannot find a disk volume specified in the current prefix for a catalog, it will ask for the correct volume to be inserted. This can be aborted by pressing Control-C.

C (Catalog- DOS 3.3 Merlin 8 only)

When you press C, the catalog of the current diskette will be shown. The Command: prompt appears to let you enter a DOS command if desired. This facility is provided primarily for locking, unlocking and deleting files. Unlike the Load Source, Save Source, and Append File commands, you must type the .S suffix when referencing a source file from this prompt. Do not use it to load or save files. If you do not want to give a disk command, just press Return. You can use Control-X to cancel a partially typed command. If you press Control-C and Return after the Command: prompt, you will be returned to the Main Menu prompt (%). You can then issue any Main Menu command such as L for Load Source. This permits you to give a Main Menu command while the catalog is still on the screen. In addition, if Control-C is pressed at the catalog pause point, printing of the remainder of the catalog is aborted.

L (Load Source)

This is used to load a source file from disk. This is a text file for ProDOS, or binary file for Merlin 8 DOS 3.3 version. You will be prompted for the name of the file. You do not have to append .S since Merlin 8/16 does this automatically. To cancel the Load Source command, just press Return and the command will be cancelled without affecting any file that may be in memory.

After a Load Source or Append Source command, you are automatically placed in the Editor. The source will automatically be loaded to the correct address.

NOTE: Subsequent Load Source or Save Source commands will display the last filename used, followed by the ? prompt. If you press Y, the current file name will be used for the command. If you press the space bar, the cursor will be placed on the first character of the filename, and you may type in the desired name. You can cancel the command by pressing Return without typing a file name. In Merlin 16, pressing the TAB key will move the cursor to the end of the default filename. Merlin 16 also lets you add a slash (/) character to the end of the name to tell Merlin not to add the .S suffix for a Load or Save operation. This is provided to save and load files to and from other editors that do not use the .S suffix.

S (Save Source)

Use this to save a source file to disk. This will save a text for ProDOS and a binary file for the DOS 3.3 version. As with the load command, you do not include the .S suffix.

NOTE: The address and length of the current source file are shown on the Main Menu, and are for information only. You do not need use these for saving; Merlin 8/16 does this automatically. As in the Load Source command above, the last loaded or saved filename will be displayed. You can press Y to save the same filename, or the space bar for a new file name. You can cancel the command by pressing Return.

A (Append File)

This loads in a specified source file and places it at the end of the file currently in memory. It operates in the same way as the Load Source command, and does not affect the default file name. It does not save the resultant combined (appended) file; you are free to do that if you wish.

D (Drive Change - DOS 3.3 Merlin 8 only)

When you press D, the drive used for saving and loading will toggle from one to two or two to one. The currently selected drive is shown on the menu. When Merlin 8 is first started, the selected drive will be the one used at startup. To change the slot number, press C for to display the current disk's catalog. Then give the disk command CATALOG,Sn, where n is the slot number. This action will catalog the newly specified drive.

See the C command for the method of changing drive specification with the ProDOS versions.

D (Disk Command - ProDOS)

This allows you to issue disk-related commands. The following commands are available:

BLOAD	pathname [,A\$....]	(only hex addresses allowed)
BRUN	pathname [,A\$....]	(only hex addresses allowed)
-	pathname [,A\$....]	(only hex addresses allowed)
BSAVE	pathname,A\$adrs,L\$len	
DELETE	pathname	
LOCK	pathname	
✓ONLINE		(shows the volumes currently on line and their names)
✓PFX	pathname	(shorthand for Prefix)
✓POP		("pops" Prefix level)
PREFIX	pathname	(sets the prefix to pathname)
RENAME	old pathname,new pathname	
SLOT	slot number	(set new slot # used by PFX= command)
UNLOCK	pathname	

A disk command returns to the disk command mode. You can then issue another disk command or just press Return to go back to the Main Menu.

Bload, Brun and -

Bload, Brun and "-" accept both BIN and SYS files. In Merlin 8, the difference between Brun and "-" is in the state of the softswitches when control is passed to the program. Brun leaves Merlin 8 up; that is, auxiliary zero page and language card RAM are selected. The "-" command switches in the

main zero page and the \$D000-\$FFFF ROMs. In Merlin 16, Brun and "-" act the same and both switch in the main zero page and the \$D000-\$FFFF ROMs. Using an RTS from such a program will return to Merlin 8/16. Most of the utility programs supplied with Merlin 8/16 such as Sourceror, XREF, etc. can be run by either method. You cannot use Brun to run programs such as the ProDOS FILER. In addition, such programs do not return to Merlin 8/16 and the /RAM/ volume is left disconnected by this procedure.

Online

This command scans all attached disk devices, including RAM disks and hard disks, and reports each volume name. This is handy for identifying the volume name of all available ProDOS disk devices.

Pfx, Prefix

If you are not sure of the volume or subdirectory pathname, you can type PFX= or PFX=1 to specify Slot 6, Drive 1, or PFX=2 for Slot 6, Drive 2.

When Prefix or Pfx is entered without a pathname, this command sets the prefix to the volume part of the current prefix. For example, if the current prefix is /MERLIN/LIB and you type Pfx and press Return at the disk command prompt, the prefix will revert to /MERLIN.

Pop

Typing POP as a disk command "pops" the directory level in the current prefix by one level. For example, if the current prefix was "/MERLIN.16/SAMPLES/GRAPHICS", typing POP would change the prefix to "/MERLIN.16/SAMPLES".

Slot

Slot is used to set the slot which *subsequent* PFX= commands will use to set the prefix from. That is to say, it does not itself set the prefix to a given slot, but rather identifies which constant slot future PFX= commands will use. For example, the following steps would set a prefix to that of the disk in a specified slot:

- 1) Press D for Disk Command, then type Online. This will help remind you know which slots, drives, and pathnames are currently active. (This is an optional step).
- 2) While still at the Disk Command prompt, type Slot n where 'n' is the desired slot and press Return.
- 3) At the Disk Command prompt type PFX=n where 'n' is the desired drive number and press Return.
- 4) Finally, press Return alone at the prompt. Note that the Prefix now shows the pathname of the specified slot and drive. Simply put, typing the Disk Commands Slot 5 then PFX= will set the prefix to slot 5.

E (Enter Editor/Assembler - Merlin 8 only)

F (Enter Full Screen Editor/Assembler - Merlin 16 only)

This command places you in the Editor/Assembler mode. It automatically sets the default tabs for the editor to those appropriate for source files.

NOTE: If you wish to use the editor to edit an ordinary (non-source) text file, you can type TABS and press Return to zero all tabs, once you are in the Editor.

O (Save Object Code)

This command is valid only after the successful assembly of a source file. In this case you will see the address and length of the object code on the menu. As with the source address, this is given for information only.

NOTE: The object address shown is the program's ORG (or \$8000, bank 0, by default) and *not that of the actual current location of the assembled code*, which is ordinarily \$8000 in auxiliary memory. When using this command, you are asked for a name for the object file. Unlike the Save Source command, no suffix will be appended to this filename.

Thus you can safely use the same name as that of the source file since the .S will not be appended to the filename. When this object code is saved to the disk its address will be the correct one, i.e., the one shown on the menu. Then when you Bload or Brun it, the program will load at that address, which can be anything (\$300, \$8000, etc).

R (Read Text File - DOS 3.3 Merlin 8 only)

This reads text files into Merlin 8. They are always appended to the current buffer. To clear the buffer and start fresh, type NEW in the editor. If no file is in memory, the name given will become the default filename. The Append command is similar, but does not change the default filename.

When the read is complete, you are placed in the Editor. If the file contains lines longer than 255 characters, these will be divided into two or more lines by the Read command. The file will be read only until it reaches HIMEM, and will produce an Out Of Memory error if it goes beyond. Only the data read to that point will remain.

The Read Text File and Write Text File commands will automatically add a T. prefix to the beginning of the filename you specify *unless* you precede the filename with a space or any other character in the ASCII range of \$20 to \$40 (%...0...9...?). This character will be ignored and not used by DOS in the actual filename. For example:

Read File:TEST will read a file called T.TEST

but,

Read File:?TEST will read a file called TEST

If you are trying to read a file called TEST and you get a File Not Found error, you can also use the Merlin 8 Catalog/Disk Command to rename the file to T.TEST.

The Read Text File and Write Text File commands are used to load or create PUT files, or to access files from other assemblers or text editors.

W (Write Text File - DOS 3.3 Merlin 8 only)

This writes a Merlin 8 source file into a text file instead of a binary file. The speed of the Read Text File and Write Text File commands is approximately that of a standard DOS Bload or Bsave. The Write Text File routine does a Verify after the write.

@ (Set Date - ProDOS)

This allows you to set the current date for ProDOS. Note that this option does not set the date on a clock card. If you have a clock, the date stamping is automatic (provided you have a Thunderclock or have installed the proper clock driver). The Set Date provision is intended for people who do not have a clock. In that case, you may use this to set the current date and this date will then be used for date stamping. You may also just use this to check on the current date. Press Return alone to exit the Set Date routine.

Q (Quit - DOS 3.3 Merlin 8 only)

This exits to BASIC. You may re-enter Merlin 8 by typing ASSEM and pressing Return. This re-entry will not destroy the source file currently in memory. This exit can be used to give disk commands, test machine language programs, run BASIC programs, etc.

Q (Quit - ProDOS)

This exits the Merlin Interpreter. If you launched Merlin 8/16, it will quit to the program that started up Merlin 8/16. If you cold started Merlin 8/16, you must specify the prefix for the next volume name and then the pathname of the next SYS file. In most cases this will be the BASIC.SYSTEM interpreter.

THE MERLIN 8 EDITOR

There are two modes in the Merlin 8 Editor: the Full Screen Editor, which includes all editing commands, and the Command Mode from which assemblies and other functions are done.

From the Merlin 8 Main Menu, you can press E to enter the Command Mode of the Editor. The Merlin 8 Command Mode is indicated by the colon (:) prompt. No actual editing is done at this level. Rather, you can either type a command which will start up the Full Screen Editor, or you may use a number of specific Command Mode operations, which are of a general utility nature, such as to print a listing, assemble a file, convert number types, etc.

When you type an editing command such as A to Add etc. from the Command Mode, the colon prompt will disappear and the screen display will change to the Full Screen Editor.

MERLIN 8 FULL SCREEN EDITOR COMMANDS

From the Merlin 8 Main Menu, press E to enter the Command Mode, then A to Add to source listing.

The current line number is shown at the upper right corner of the screen. To the left of the line number is a vertical bar which is the End-of-Line Marker. It indicates the position at which an assembly listing will overflow the printer line. You can put characters beyond this mark, but they should be for information only, and will not be printed within a printer listing.

The fields are tabbed, and the arrow keys can be used to move the cursor to the next tab position.

The Edit Mode commands are divided into two types: Control key commands which are line oriented, and Open-Apple key commands which are global, i.e. oriented to the entire listing. The control key commands edit text and move the cursor on just the line the cursor is presently on. Use the Open-Apple key commands to make changes to groups of lines, or to move about in the listing. All editing commands work whether the Full Screen Editor was started up using the Add, Insert or Edit commands. When you are through editing, press Open-Apple-Q. The line is accepted as it appears on the screen, no matter where the cursor is when you exit the Edit Mode.

To get the most out of the Merlin 8 Full Screen Editor, you should keep in mind that a full screen editor is like a word processor. That is, any character you type is immediately entered into whatever line the cursor is on.

With the Merlin 8 Full Screen Editor, if you can see it on the screen, you can edit it, and moving to a line is a simple matter of using the arrow keys or other special commands to move to the part of the listing you want to edit. Just remember, when you are using the Full Screen Editor, think of yourself as using a word processor where you can freely scroll to whatever part of the page you want to edit, and the final document is just your source listing.

MERLIN 8 CONTROL KEY COMMANDS (Line oriented)

Control-B (Beginning of line)

Moves the cursor to the beginning of the line.

Control-D (Delete character)

Deletes the character under the cursor. Also see Delete.

Control-F (Find)

Finds the next occurrence on the current line of the character typed after the Control-F. The cursor changes to an inverse F to indicate the Find Mode. To move the cursor to the next occurrence on the line, press the desired character key again. Typing any other character will exit the Find Mode and enter the text typed at that position.

Control-I or TAB (Toggle insert cursor)

Toggles the cursor mode between the insert cursor designated by an inverse I, and overstrike cursor which appears as an inverse block. The insert mode of the cursor should not be confused with entering the Full Screen Editor using the Add and Insert commands. The cursor can be in the insert mode regardless of whether lines are being added or inserted. The insert mode of the cursor refers only to whether individual characters are being inserted or typed over.

The character insert mode defaults to ON upon entry. When you change it with Control-I, it remains that way until changed again. Thus, moving from one line to another has no effect on this status.

Control-L (Lower case convert)

The Merlin 8/16 PARMS file can be configured so that unless the cursor is in a comment or an ASCII string, lower case characters will be converted to UPPER CASE characters. To override this conversion, or to reinstate it, just use the Control-L command. This conversion is also in effect when you use the Open-Apple-F, Open-Apple-W, or Open-Apple-L find commands to specify the text to find. Even if enabled in the PARMS file, this conversion is defeated when the tabs are zeroed.

Control-N (End of line)

Moves the cursor to the end of the line.

Control-O (Other characters)

This is used as a special prefix key from the Command Mode. For example, if you wanted to type a Control-I or an Escape as part of a PRTR initialization string, you would press Control-O, followed by the control character you desire. The control character will appear in inverse. For multiple control characters, Control-O must be typed before each character is entered.

Control-R (Restore)

This command restores the original line. For example, if you have used Control-Y to delete all characters to the end of the line, you can press Control-R to undo the effects of the Control-Y command.

Control-S (Status box)

This command displays a status box showing the number free and used bytes.

Control-T (Set marker on current line - Vertical "Tab")

This command can be used to set a marker at the current line for recall by the Open-Apple-T command.

Control-W (Find word)

This command moves the cursor to the beginning of each word in the line (alphanumeric).

Control-X (Exit global exchange, etc.)

This command can be used to cancel any global exchange, text selection, or string search while it is in progress.

Control-Y (Delete to end of line)

Deletes all characters from the cursor to the end of the line.

Arrow keys (Cursor movement)

The arrow keys move the cursor in the specified direction.

Delete (Delete character)

Deletes the character to the left of the cursor. Also see Control-D.

Escape (Move to beginning of next line)

This command moves the cursor to the beginning of the next line. This is similar to Return except that Escape does not insert a blank line.

Return (Insert blank line)

Pressing Return anywhere in the line causes the cursor to move to the beginning of the next line and insert a blank line.

TAB (Toggle insert cursor)

Toggles the cursor mode between the insert cursor, which is shown as an inverse I, and the overstrike cursor, which is shown as an inverse block.

Moving from one line to another has no effect on the status of the cursor; it only changes when toggled with TAB. Also see Control-I.

MERLIN 8 OPEN APPLE KEY COMMANDS (Entire listing oriented)

In addition to the line-oriented commands (control key commands), the Merlin 8 Full Screen Editor uses Open-Apple (⌘) key commands to move within the listing, and to edit entire lines of text. These commands are as follows:

⌘B (Beginning of source)

This command moves the cursor to the beginning of the source listing.

⌘D (Delete current line)

This command deletes the current line and places it in a special 'undo' buffer which is independent of the clipboard.

The ⌘R command replaces the current line with the contents of the 'undo' buffer. Therefore, to move a single line to another location, you could place the cursor on the line to be moved, and then type ⌘D to delete the line. Then move the cursor to another line, press Return, ⌘I or ⌘Tab to create an empty line, and press ⌘R to replace that line with the deleted line. Also see ⌘Delete.

⌘E (Global Exchange, also called 'Find & Replace')

Sometimes called 'Find & Replace,' this command will let you search for a group of words, and replace them with another. The ⌘E command opens a dialog box that asks for the text to change, and the new text to replace it. If you press Return alone for either of these, the command is canceled.

If you enter the text in both fields and press Return, the file is then searched for the change text. Unlike the FIND command, it looks only for full words. That is, the text found must be bounded by non-alphanumeric characters or it will be ignored.

If text is found with this method, the screen is reprinted with the replacement made and the cursor is placed on the first character of the replacement. Now you must press a key to continue. Pressing Return (or most any other control character) will defeat the change and the command will look for the next occurrence of the text to change. Pressing the space bar or any other character, except A, will accept the change and the routine will continue.

You can back out of the global exchange while the cursor is on an entry by pressing Control-X. You can also press the A key, which will cause *all* occurrences to be changed.

You can tell when the routine is finished by the fact that during the exchange sequence, the line number at the top right is missing. The line number will return when there are no more matches for the change text, or when you press Control-X.

⌘F (Find text)

The ⌘F command opens a window which asks for the text to find. It then finds the first occurrence of the text in the entire text file. The text can be anywhere on a line. After the first find, you can find the next occurrence by typing another ⌘F. You can edit the line, and then type ⌘F to go to the next occurrence.

During the search function, one or more plus (+) signs will be shown next to the line number at the top right of the screen. This is only an indicator that the search function is active. The number of plus signs shown is arbitrary, and has no relation to the total number of occurrences.

In Merlin 16, if the ⌘F command is used after text has been selected, only the selected text will be searched for the text to be found. When the search has been completed, the text is no longer selected. Thus, you can use the ⌘Y, ⌘C or ⌘X commands to search just a portion of your listing.

The ⌘B command and the Control-S status command both cancel the Find mode, as does failure to find the text below the current line.

The ⌘W command is identical to ⌘F except that it finds only whole words bounded by non-alphanumeric characters. If you type either ⌘W or ⌘F to find the next occurrence, this mode will change accordingly.

In all cases the line containing the text is moved to the center of the screen, unless it is within the first 10 lines of the start of the source.

⌘I (Insert line)

Pressing ⌘I will insert a blank line at the cursor. Also see ⌘TAB.

⌘L (Locate label, marker or line number)

This command will locate the first occurrence of a label or any text in the label column. Only the characters typed are compared with the labels. Thus the search string LOOP would jump to the label LOOP2 if LOOP did not occur first. To find a specific label when there may be another similar label, end the input with a single space.

If a number is entered after this command, the cursor will move to the beginning of the line number specified. This is particularly handy when editing a source file from a printed listing.

The intended use for this command is to move rapidly to a particular place in the source. You can use create your own 'markers' to enhance the capability of this command. Thus, if a line has *7 in it, you can specify *7 as the text to find for this command and it will locate it.

In all cases the line containing the text is moved to the center of the screen, unless it is within the first 10 lines of the start of the source.

␣N (End of source)

This command moves the cursor to the end of the source listing.

␣Q (Quit full screen editor)

This command returns to the Command Mode.

␣R (Replace)

This command exchanges the current line with the contents of the 'undo' buffer. Therefore, pressing ␣R a second time will cancel the effect of the first press.

Using ␣R when the cursor is on blank line will place the contents of the 'undo' buffer on the line and place the empty line in the 'undo' buffer.

The ␣R command can be used to move a single line. Place the cursor at the beginning of the line to be moved and press ␣R. Move the cursor to the desired location, press Return to insert a blank line, and press ␣R again.

␣R can be used by itself to easily interchange two lines. Just place the cursor on the first line, press ␣R, move the cursor to the second line and press ␣R again. Then move the cursor back to where the first line was and press ␣R for the third, and final time. Also see ␣D.

␣T (Return to line marker (vertical "Tab"))

This command returns to the line marked by the last Control-T command.

⌘V (Paste)

Pastes the contents of the clipboard at the line containing the cursor. Only full lines are moved. Using this command does not change the contents of the clipboard, so this command can be used to replicate a range of lines.

If the ⌘V paste command is issued when a range of text has been selected, the text on the clipboard will be inserted before the last line of selected text.

⌘W (Find word)

The ⌘W command is identical to ⌘F except that it finds only whole words bounded by non-alphanumeric characters. If you type either ⌘W or ⌘F to find the next occurrence, this mode will change accordingly.

If the ⌘W command is used after text has been selected, only the selected text will be searched for the word to be found. When the search has been completed, the text is no longer selected. The search can be cancelled with Control-X.

⌘X (Cut to Clipboard)

⌘X starts the select mode to cut text. The first time ⌘X is pressed, the current line is selected and is shown in inverse. Use the down arrow or Escape keys to extend the selection if desired, or press any other key to cancel the selection. Additional selected lines are shown in inverse. Use the up arrow key to adjust the range selected if you go too far. The select mode will be canceled if you move the cursor above the first selected line or *past the top of the current screen*.

The second time ⌘X is pressed the selected text is cut from the listing and is placed on the clipboard. Merlin 8 has no "copy" command, but if you immediately follow the ⌘X command that cuts a portion of text with ⌘V (for paste), the desired text will be on the clipboard, and the original listing will be restored.

If you are unfamiliar with the idea of a "clipboard", this is just an analogy to how you might put a piece of paper clipped from a magazine, letter, etc. on a clipboard, to hold it temporarily while you were getting ready to put it in its final location. In Merlin 8, the clipboard refers to a memory buffer that holds the text you have cut while you decide where you want the final text placed. Using the clipboard, you can cut text from one source file, load another, and then paste the text into a second file. The clipboard is cleared when a file is assembled.

␣Y (Select all text)

This command selects all text to be cut from the current line to the end of the listing. ␣X will cut the text, while pressing any other key will cancel the selection. This technique can be used to move the entire listing to the clipboard.

␣Z (Center screen)

This command repositions the screen so that the line the cursor is on becomes the center line on the screen.

␣Delete (Delete)

This command deletes the line *above* the cursor and places it in a special 'undo' buffer which is independent of the clipboard.

The ␣R command replaces the current line with the contents of the 'undo' buffer. Therefore, you could use ␣Delete to delete a line, move the cursor to another line, press Return, ␣I or ␣Tab to insert a line, and press ␣R to replace that line with the deleted line.

␣TAB (Insert line)

Pressing ␣TAB will insert a blank line at the cursor. Also see ␣I

␣Down arrow (Move half-screen down)

Moves the cursor down 10 lines; that line then becomes the center line on the screen. This command has the effect of moving the current line to the top of the screen and then moving the cursor to what was the bottom line on the screen.

␣Up arrow (move half-screen up)

Moves the cursor up 10 lines; that line then becomes the center line on the screen. This command has the effect of moving the current line to the bottom of the screen and then moving the cursor to what was the 1st line on the screen.

⌘8 (Asterisk)

Produces a line of 32 asterisks. Overstrikes existing line, if any. Undo with Control-R.

⌘9 (Box)

Produces an asterisk, 30 spaces, and then another asterisk. This and the ⌘8 command can be used to produce a large box for titles and other information. Overstrikes existing line, if any. Undo with Control-R.

⌘- (Hyphen)

Produces a line of 1 asterisk and 31 hyphens. Overstrikes existing line, if any. Undo with Control-R.

⌘= (Equal sign)

Produces a line of 1 asterisk and 31 equal signs. Overstrikes existing line, if any. Undo with Control-R.

MERLIN 8 EDITOR COMMAND SUMMARY

CONTROL KEY COMMANDS (line oriented)

The Control Key commands consist of cursor moves and line oriented commands.

Control-B	Moves cursor to beginning of line
Control-C	Cancel assembly and return to Command Mode
Control-D	Deletes character under the cursor
Control-F	Finds next occurrence of next character typed
Control-I	Toggles insert and overstrike cursor
Control-L	Toggles lower case conversion
Control-N	Moves cursor to end of line
Control-O	Prefix key for typing optional characters
Control-R	Restores original line
Control-S	Displays status box
Control-T	Set marker on current line for recall by CT (vertical "Tab")
Control-W.....	Finds next occurrence of word in line
Control-X.....	Exits global exchange, etc. while in progress
Control-Y	Delete characters to end of line
Arrows	Moves the cursor in the specified direction
Delete	Deletes character to left of cursor
Escape	Moves cursor to beginning of next line
Return	Moves cursor down and inserts blank line
TAB	Toggles insert and overstrike cursor

OPEN-APPLE KEY COMMANDS (entire listing oriented)

The Open-Apple Key commands are global commands, which means they are generally oriented to the whole listing as opposed to just the current line (or a single character).

⌘B	Moves to beginning. Cursor on center line
⌘D	Deletes line and places it in 'undo' buffer
⌘E	Global Exchange (Search & Replace)
⌘F	Finds next occurrence of text entered
⌘I	Inserts blank line at cursor
⌘L	Finds first occurrence of label or line
⌘N	Moves cursor to end of listing
⌘Q	Returns editor to Command Mode
⌘R	Swaps current line with 'undo' buffer
⌘T	Goes to line of last Control-T (go to vertical "Tab")
⌘V	Pastes contents of clipboard on current line
⌘W	Finds next occurrence of whole word
⌘X	Start text selection/Cut selected text to clipboard
⌘Y	Selects text from current line to end of file
⌘Z	Current line becomes center line on screen
⌘Delete	Deletes line above cursor; puts in 'undo' buffer
⌘TAB	Inserts a blank line at cursor
⌘Down	Moves cursor down 10 lines
⌘Up	Moves cursor up 10 lines
⌘8	Produces a line of 32 asterisks
⌘9	Produces 1 asterisk, 30 spaces, and 1 asterisk
⌘-	Produces a line of 1 asterisk followed by 31 hyphens
⌘=	Produces a line of 1 asterisk followed by 31 equal signs

MERLIN 8 GENERAL REMARKS

When you move the cursor between lines, its horizontal position will jump around. This is because it is based on the actual position in the line and not on the screen position. If the tabs are zeroed you will not notice this, except for the fact that the cursor is never beyond the last character in the line.

The maximum line length is 80 characters. Lines longer than that will be truncated IF they are edited.

You must return to the Command Mode (⌘Escape) in order to use the ASM command to assemble, MON to use the Merlin 8 Monitor, or to Quit and go to the Main Menu, etc. An assembly will delete the contents of the clipboard.

OOPS

Virtually any Editor action can be undone. You should remember that the proper undo command is of the same 'type' as the command you want to undo. Thus, any Control key command is undone by Control-R. This includes the $\text{C}8$, $\text{C}9$, $\text{C}-$, and $\text{C}=\text{}$ commands which are considered line oriented commands for this purpose.

The line deletion commands $\text{C}D$ and $\text{C}Delete$ are undone by creating an empty line with $\text{C}Tab$ followed by $\text{C}R$. If you forget to create the empty line, type another $\text{C}R$ and then insert the empty line to receive the undo buffer contents.

The $\text{C}R$ command undoes itself.

A Cut ($\text{C}X$) is undone by a Paste ($\text{C}V$) without moving the cursor off its line.

If you are entering a line of text in response to a prompt, such as a filename, PRTR initialization, or dialog box, you can press Control-C or Control-X to cancel the line.

ED.16

The full screen editor in Merlin 8 can be replaced with a version written specifically for the 65802 or 65816 chip. This editor is named ED.16 on both DOS 3.3 and ProDOS Merlin 8 disks. To enable ED.16 on the DOS 3.3 disk, change line 80 of the HELLO program to BRUN ED.16 instead of ED as is now done. On the ProDOS disk, you will have to rename ED to ED.OLD, and rename ED.16 to ED. Although ED.16 is provided on the Merlin 8 disks, it is unlikely you will ever need it. We recommend that if you do have the 65802 or 65816 that you use Merlin 16 instead of Merlin 8.

EDMAC and KEYMAC

These are macro utilities that will automatically type entire phrases or lines in your source file with a single keystroke. EDMAC is for use with the full screen editor, KEYMAC is for use with the line editor (presuming that ED has not been loaded). See the Utilities section of this manual for a description of the actual keyboard definitions in these files. EDMAC and KEYMAC can be activated within Merlin 8 by typing BRUN EDMAC or BRUN KEYMAC from the Main Menu, as appropriate.

Editor Technical Information

The ED and ED.16 files have been arranged so that certain parameters can be changed with a little effort. At relative byte 3 in the file there is an address that points to the main part of the program, hereinafter referred to as START, that is past a relocating header. At START+10 is a table of command characters used without the C key. This table ends in a zero. Following this is the table of the key commands used with the C , again ending in zero. If you are inclined, you can change these definitions to change the command keys of the editor.

MERLIN 8 COMMAND MODE

GENERAL GUIDELINES FOR THE COMMAND MODE

The Command Mode is used for assemblies, printing the source listing, and other general-purpose functions. The command mode is entered by pressing E at the Main Menu, or by pressing $\text{C}Q$ while in the Full Screen Editor.

For most of the Merlin 8 Command Mode operations, only the first letter of the command is required, the rest being optional. This manual will show the required command characters in UPPER case and the optional ones in lower case.

ABOUT THE MERLIN 8 COMMAND MODE DOCUMENTATION

For each of the commands available in the Merlin 8 Command Mode, the documentation consists of three basic parts:

- 1) the name and syntax of the command
- 2) examples of the use of each available syntax
- 3) a description of the function of each command

When the syntax for each command is given:

PARENTHESES () indicate a required value

ANGLE BRACKETS <> indicate an optional value or character

SQUARE BRACKETS [] are used to enclose comments about the command

Line Numbers in Command Mode

With some commands, you must specify a line number, a range of line numbers, or a range list. A line number is just a number. A range is a pair of line numbers separated by a comma. A range list consists of several ranges separated by a slash (/).

Line Number examples:

10	LINE #	[a single line number]
10,30	RANGE	[the range of lines 10 to 30]
10,30/50,60	RANGE LIST	[ranges 10 to 30 AND 50 to 60]

If a line number in a range exceeds the number of the last line in the source, the editor automatically adjusts the specified line to the last line number. For example, if you wanted to Delete all the lines past 100 in a source listing, entering D100,9999 would do it.

Delimited Strings (or d-strings)

Several commands allow specification of a string. The string must be delimited by a non-numeric character other than the slash or comma. Such a string is called a delimited or d-string. The usual delimiter is single or double quote marks (' or ").

Delimited string examples:

```
'this is a delimited string'  
"this is a delimited string"  
@this is another d-string@
```

Note that the slash (/) cannot be used as a delimiter since it is the character that delimits range lists in the Editor.

Wild Card Characters in Delimited Strings

For all of the commands that use delimited strings or d-strings, the '^' character acts as a wild card character. For example, the d-string 'Jon^s' is equivalent to both 'Jones' and 'Jonas' d-strings.

Upper and Lower Case Control

The shift and caps lock keys work as you would expect. While editing or entering a line of text, you can also use the Control-L command, described earlier in this section.

COMMAND MODE COMMANDS

Following are the commands recognized by Merlin 8 in the Command Mode of the Editor. The Command Mode is indicated by the colon prompt (:).

A (Add)

A [only option for this command]

The Add command places you in the Full Screen Editor at the end of the existing source listing, if any. Adding lines is much like entering additional BASIC lines with auto line numbering. To exit from ADD mode, press C^{O} .

You may enter an empty line by pressing space then Return. This is useful for visually blocking off different parts of a listing.

E (Edit)

Edit (line number)

E 10

[edits line number 10]

This enters the full screen editor, with the cursor on the specified line.

C (Change)

Change (d-string d-string)

Change (line number) <d-string d-string>

Change (range) <d-string d-string>

Change (range list) <d-string d-string>

C "hello"goodbye [finds "hello" and if told to do so will change it to "goodbye"]

C 50 "hello"bye [changes in line 50 only]

C 50,100 "Hello"BYE [changes lines 50 through 100]

C 50,60/65,66 "AND"OR [changes in lines 50 through 60 and lines 65 and 66]

This changes occurrences of the first string to the second string. The strings must have the same delimiters. For example, to change occurrences of *speling* to *spelling* throughout the range 20,100, you would type C 20,100 "speling"spelling. If no range is specified the entire source file is used.

Before the change operation begins, you are asked whether you want to change *all* or *some*. If you select *some* by pressing the S key, the editor stops whenever the first string is found and displays the line as it would appear with the change.

If you then press the Y key, the change will be made. If you press Return, the change will not be made. Typing any control character such as Escape, Return or any others will result in the change not being made. Any other key, such as Y or even N, will accept the change. Control-C or the slash (/) key will abort the change process.

COPY

COPY (line number) TO (line number)

COPY (range) TO (line number)

COPY 10 TO 20 [copies line 10 to just before line 20]

COPY 10,20 TO 30 [copies lines 10 through 20 to just before line 30]

This copies the line number or range to just *above* the specified number. It does not delete anything.

CW (Change word)

Change (d-string d-string)

Change (line numbers) <d-string d-string>

Change (range) <d-string d-string>

Change (range list) <d-string d-string>

CW "PTR" "PRT" [change all "PTR"s to "PRT"s]

CW 20 "PTR" "PRT" [as above but only in line 20]

CW 20,30 "PTR" "PRT" [do the same as the above but for lines 20 through 30]

CW 1,9/20,30 "PTR" "PRT" [same as above but include lines 1 through 9 in the range]

This works similar to the CHANGE command with the added features as described under Find Word (FW).

D (Delete)

Delete (line number)

Delete (range)

Delete (range list)

D 10 [deletes line number 10]

D 10,32 [deletes lines 10 through 32]

D 20,30/10,12 [deletes ranges of lines 10 through 12 and 20 through 30]

This deletes the specified lines. Unlike BASIC, the line numbers are fictitious; they change with any insertion or deletion.

NOTE: When deleting several blocks of lines at the same time, you *must* specify the higher range first for the correct lines to be deleted.

F (Find)

Find (d-string)

Find (line number) <d-string>

Find (range) <d-string>

Find (range list) <d-string>

F "A String" [finds lines with "A String"]

F 10 "STRING" [finds "STRING" if in line 10]

F 10,20 "HI" [finds lines in range of 10 through 20 that contain "HI"]

F 10,20/50,99 "HI" [finds lines that contain "HI" in range of 10 through 20 and 50 through 99]

FIX

FIX [only option for this command]

This undoes the effect of the TEXT command. It also does a number of technical housekeeping chores. It is recommended that FIX be used on all source files from external sources that are being converted to Merlin 8 source files, after which the file should be saved.

NOTE: The TEXT and FIX commands are somewhat slow. Several minutes may be needed for their execution on large files. FIX will truncate any lines longer than 255 characters.

FW (Find Word)

FW (d-string)

FW (line number) <d-string>

FW (range) <d-string>

FW (range list) <d-string>

FW "LABEL"

[find all lines with "LABEL"]

FW 20 "LABEL"

[try to find "LABEL" in 20]

FW 20,30 "PTR"

[find all lines between 20 and 30 that contain "PTR"]

FW 20,30/50,99 "PTR"

[find all lines between 20 and 30 and between 50 and 99 that contain the word "PTR"]

This is an alternative to the FIND command. It will find the specified word only if it is surrounded, in source, by non-alphanumeric characters.

Therefore, FW "CAT" will find:

CAT

CAT-1

(CAT,X)

but will not find CATALOG or SCAT.

GET

GET (obj adrs)

GET

[put object code in Main Memory at the address specified in the source's ORG]

GET \$4000

[put object code at location \$4000 in Main Memory]

This command is used to move the object code, after an assembly, from its location in Auxiliary Memory to its ORG location in Main Memory. The address must be above the existing source file, if any, and it will not be allowed to overwrite DOS. You can do a NEW if you want to load it lower in memory than allowed, but you must remember to save the source first. You cannot use GET to put object code at memory locations lower than \$901, but you can go to the Monitor afterwards and use it to move the object to any desired location. However, any such move using the Monitor may destroy your source or other data necessary to Merlin 8's operation. Caution should be used.

The GET command does not check if a valid object code has been assembled.

This command is supplied for convenience only. The recommended method for testing a program is to save the source code first, save the object code, and then run the program from BASIC or with the G command from the Monitor.

HEX-DEC CONVERSION

128 = \$0080

\$80 = 128

If you type a positive or negative decimal number in the Command Mode, the hex equivalent is returned. If you type a hex number using the \$ prefix, the decimal equivalent is returned. All commands accept hex numbers.

L (List)

List

List (line number)

List (range)

List (range list)

L

[list entire file]

L 20

[list line 20 only]

L 20,30

[list 20 through 30]

L 20,30/40,42

[list 20 through 30 and then list lines 40 through 42]

Lists the source file with line numbers. Control characters in source are shown in inverse, unless the listing is being sent to a printer or other non-standard output device.

The listing can be aborted by Control-C or with the slash (/) key. You may stop the listing by pressing space and then advance a line at a time by pressing space again. By holding down space, the auto-repeat feature of the Apple will result in a slower listing. Any other key will resume the normal speed. This space bar technique also works during assembly and the symbol table printout. Any other key will restart it. The space bar pause also works during assembly and the symbol table printout.

LEN (Length)

LEN [only option for this command]

This gives the length in bytes of the source file, and the number of bytes free.

MON (Monitor)

MON [only option with this command]

This exits to the Monitor. You may return to the Merlin 8 Main Menu by pressing Control-C, Control-B, or Control-Y. These commands re-establish important zero page pointers from a safe area inside Merlin 8. Thus Control-Y will give a correct entry even if you have damaged the zero page pointers while in the Monitor. DOS is not connected while using this entry to the Monitor.

You may return to the editor directly by typing 0G and pressing Return but unlike the above commands, this uses the zero page pointers stored at \$0A-\$0F. Therefore, you must be sure that these pointers have not been altered. For normal usage, any of the three Control commands should be used to return to Merlin 8.

When you exit to the Monitor with the MON command, the RAM-based \$D000-\$FFFF memory is enabled, and therefore, Merlin 8 and it's symbol table if any. If you want to examine the ROM memory that would normally correspond to Applesoft and the F8 Monitor, you should quit Merlin 8 with the Main Menu Quit command, and enter the Monitor with Call -151. Under ProDOS, this procedure necessitates loading BASIC.SYSTEM which removes Merlin 8 from memory.

MOVE

MOVE (line number) TO (line number)

MOVE (range) TO (line number)

MOVE 10 TO 20 [Move line 10 to just before 20]

MOVE 10,20 TO 30 [Move lines 10 through 20 to just before line 30]

This is the same as COPY but after copying, automatically deletes the original range.

NEW

NEW [only option for this command]

Deletes the present source file in memory.

P (Print without format)

Print

Print (line number)

Print (range)

Print (range list)

P	[print entire file]
P50	[print line 50 only]
P50,100	[print lines 50 through 100]
P1,10/20,30	[print 1 through 10 and then print lines 20 through 30]

This is the same as LIST except that line numbers are not added. See PRTR for formatted printouts.

PRTR (Formatted printout)

PRTR (command)

PRTR 1	[activate printer in slot 1 with no printer init string]
PRTR 1 ""Page Title"	[printer in slot 1, no printer init string, "Page Title" is the page header]
PRTR 1 "<Control-I>80N"	[as above, add Control-I80N to initialize the printer]
PRTR 3	[send formatted listing to screen]

This command is for sending a listing to a printer with page headers and provision for page boundary skips. No header is printed on the first page of the printout. See the section on Configuration for details on setting up default parameters, also TTL in the section on The Assembler.

The entire syntax of this command is:

PRTR # "(string)"<page header>"

If the page header is omitted, the header will consist of page numbers only.

The initialization string may not be omitted if a page header is to be used. If no special string is required by the printer, use a null string of two quotes only (""), as in the example showing "Page Title" in which case a carriage return will be used.

No output is sent to the printer until a LIST, PRINT, or ASM command is issued. See Control-O for information on inserting Control characters in the printer init string. The PRTR command only affects the next output command, and is canceled at the end of the listing.

Q (Quit)

Q [only option for this command]

Exits to Main Menu.

TABS

TABS <number><, number><,...> <'tab character'>
TABS [clear all tabs]
TABS 10,20 [set tabs to 10 & 20]
TABS 10,20 " " [as above, space is tab character]

This sets the tabs for the editor, and has no effect on the assembler listing. Up to nine tabs are possible. The default tab character is a space, but any may be specified. The assembler regards the space as the only acceptable tab character for the separation of labels, opcodes, and operands. If you don't specify the tab character, then the last one used remains. Entering TABS and a Return will set all tabs to zero.

TEXT

TEXT [only option for this command]

This converts all spaces in a source file to inverse spaces. The purpose of this is for use on word processing type text files so that it is not necessary to remember to zero the tabs before printing such a file. This conversion has no effect on anything except the Editor's tabulation. The command FIX undoes the effect of the TEXT command.

TROF (Truncate Off)

TROF [only option for this command]

When used as an Immediate command, returns to the default condition of the truncation flag which also happens automatically upon entry to the editor from the Main Menu or from the Assembler. All source lines when listed or printed will appear normal.

TRON (Truncate On)

TRON [only option for this command]

When used as an Command Mode command, sets a flag which, during LIST or PRINT, will suppress printing of comments that follow a semicolon. It makes reading of some source files easier.

USER

USER
 USER 1 [example for use with XREF]
 USER 0: FILENAME [example for use with PRINTFILER]

This does a JSR to the routine at \$3F5. This is the location of the Applesoft ampersand vector which normally points to a RTS. USER is designed to connect the various utilities supplied with Merlin 8 and for user defined printer drivers. You must be careful that your printer driver does not use zero page addresses, with the exception of the I/O pointers and \$60-\$6F, because this is likely to interfere with Merlin 8's heavy usage of zero page. Several supplied utilities operate through the USER command.

VAL

VAL "expression"
 VAL "PTR" [return value of label "PTR"]
 VAL "LABEL" [gives the address (or value) of LABEL for the last assembly done or "unknown label" if not found.]
 VAL "\$1000/2" [returns \$0800]
 VAL "%1000" [returns \$0008]

This will return the value of the expression as the assembler would compute it. All forms of label and literal expressions valid for the assembler are valid for this command. Note that labels will have the value given them in the most recent assembly.

VID (VIDeo)

VID (slot number)
 VID 3 [turns on 80 column display]

This command selects an 80 column display device. To turn off the display, use Escape Ctrl-Q for the Apple IIe, IIc or IIgs video; use Escape-0 for the Videx UltraTerm. VID 3 is required to re-activate the display if you have set the PARMs file to switch to 40 columns for the PRTR command.

W (Where)

Where (line number)

W 50

[where is line 50 in memory]

W 0

[where is end of source file]

This prints in hex the location in memory of the start of the specified line. *Where 0* or *W0* will give the location of the end of source.

. [period]

.

[only option for this command]

Lists starting from the beginning of the last specified range. For example, if you type *L10,100*, lines 10 to 100 will be listed. If then you use the period (.) command, the listing will start again at 10 and continue until stopped. The end of the range is not remembered.

/ (slash)

/ <line number>

/

[start to list at last line listed]

/50

[start listing at line 50]

This command continues the listing from the last line number listed, or, when a line number is specified, from that line. This listing continues to the end of the file or until it is stopped as in LIST.

ASSEMBLING A MERLIN 8 FILE

Once you have entered and edited your source listing, you will want to assemble it. ASM does that. Remember to exit the Full Screen Editor by pressing C Escape. When you see the colon (:) prompt, type ASM and press Return.

ASM (Assemble)

ASM [only option for this command]

This passes control to the assembler, which attempts to assemble the source file.

If you wish to have a formatted printed listing of an assembly, just use the PRTR command immediately before typing in the ASM command.

Control-C (Cancel assembly)

Terminates assembly and returns to the Command Mode.

Control-D (Toggle Display Status)

Control-D [only option for this command]

During the second pass of assembly, pressing Control-D will toggle the list flag, so that the listing will either stop or resume. The next LST opcode in the source overrides this, but another Control-D can be used again.

THE MERLIN 16 EDITOR

From the Merlin 16 Main Menu, you can press F to enter the Full Screen Editor, or if you have loaded a source file, you will enter the Full Screen Editor automatically.

At the upper right hand corner of the screen, the number of the line on which the cursor is located is shown. Somewhat to the left of this you may see a vertical bar. This bar is the End-of-Line Marker and it indicates the position at which an assembly listing will overflow the printer line. You can put characters beyond this mark, but they should be for information only, and will not be printed within a printer listing.

The fields are tabbed, and the arrow keys can be used to move the cursor to the next tab position.

The commands are divided into two types: Control key commands which are line oriented, and Open-Apple key commands which are global and thus oriented to the entire listing. The Control key commands edit text and move the cursor on just the line the cursor is presently on. Use the Open-Apple key commands to make changes to groups of lines, or to move about in the listing. When you are through editing, the line is accepted as it appears on the screen, no matter where the cursor is. The file is assembled by pressing ⌘A; ⌘Q will return you to the Main Menu. Other commands are available in the Command Box (⌘O).

To get the most out of the Merlin 16 Full Screen Editor, you should keep in mind that a full screen editor is like a word processor. That is, any character you type is immediately entered into whatever line the cursor is on.

With the Merlin 16 Full Screen Editor, if you can see it on the screen, you can edit it, and moving to a line is a simple matter of using the arrow keys or other special commands to move to the part of the listing you want to edit. Just remember, when you are using the Full Screen Editor, think of yourself as using a word processor where you can freely scroll to whatever part of the page you want to edit, and the final 'document' is just your source listing.

MERLIN 16 CONTROL KEY COMMANDS (Line oriented)

Control-B (Beginning of line)

Moves the cursor to the beginning of the line.

Control-C (Cancel assembly)

Cancels assembly and returns to the Editor with cursor on line where assembly was interrupted.

Control-D (Delete character)

Deletes the character *under* the cursor. Also see Delete.

Control-F (Find)

Finds the next occurrence on the current line of the character typed after the Control-F. The cursor changes to an inverse F to indicate the Find Mode. To move the cursor to the next occurrence on the line, press the desired character key again. Typing any other character will exit the Find Mode and enter the text typed at that position.

Control-I or TAB (Toggle insert cursor)

Toggles the cursor mode between the insert cursor (inverse I) and overstrike cursor (inverse block). The cursor can be in the insert mode regardless of whether lines are being added or inserted. The insert mode of the cursor refers only to whether individual characters are being inserted (inverse I) or typed over (inverse block).

The character insert mode defaults to ON upon entry. When you change it with Control-I, it remains that way until changed again. Thus, moving from one line to another has no effect on this status.

Control-L (Lower case convert)

The Merlin 16 PARMS file can be configured so that, unless the cursor is in a comment or an ASCII string, lower case characters will be converted to UPPER CASE characters. To override this conversion, or to reinstate it, just use the Control-L command. This conversion is also in effect when you use the Open-Apple-F, Open-Apple-W, or Open-Apple-L find commands to specify the text to find. Even if configured in the PARMS file, this conversion is defeated when the tabs are zeroed.

Control-N (End of line)

Moves the cursor to the end of the line.

Control-O (Other characters)

This is used as a special prefix key. For example, if you wanted to type a Control-I or an Escape as part of an ASC string, you would press Control-O, followed by the control character you desire. For multiple control characters, Control-O must be typed before each character is entered.

Control-R (Restore)

This command restores the original line. For example, if you have used Control-Y to delete all characters to the end of the line, you can press Control-R to undo the effects of the Control-Y command.

Control-S (Status box)

This command displays a status box showing the number free and used bytes in the current source listing workspace.

Control-T (Set marker on current line - Vertical "Tab")

This command can be used to set a marker at the current line for recall by the Open-Apple-T command.

Control-W (Find word)

This command moves the cursor to the beginning of each word in the line (alphanumeric).

Control-X (Cancel global Exchange, etc.)

This command can be used to cancel any global exchange, text selection or string search while it is in progress.

Control-Y (Delete to end of line)

Deletes all characters from the cursor to the end of the line.

Arrow keys (Cursor movement)

The arrow keys move the cursor in the specified direction.

Delete (Delete character)

Deletes the character to the *left* of the cursor. Also see Control-D.

Escape (Move to beginning of next line)

This command moves the cursor to the beginning of the next line. This is similar to Return except that Escape does not insert a blank line.

Return (Insert blank line)

Pressing Return anywhere in the line causes the cursor to move to the beginning of the next line and insert a blank line.

TAB (Toggle insert cursor)

Toggles the cursor mode between the insert cursor (inverse I) and overstrike cursor (inverse block).

Moving from one line to another has no effect on the status of the cursor; it only changes when toggled with TAB. Also see Control-I.

MERLIN 16 OPEN APPLE KEY COMMANDS (Entire listing oriented)

In addition to the line-oriented commands (control key commands), the Merlin 16 Full Screen Editor uses Open-Apple (⌘) key commands to move within the listing, and to edit entire lines of text. These commands are as follows:

⌘A (Assemble)

This command passes control to the assembler which attempts to assemble the source file.

⌘B (Beginning of source)

This command moves to the beginning of the source listing, cursor on line 1.

⌘C (Copy)

⌘C starts the select mode to copy text. The first time ⌘C is pressed, the current line is selected and is shown in inverse. Use the down arrow or Escape keys to extend the selection if desired, or press any other key to cancel the selection. Additional selected lines are shown in inverse. Use the up arrow key to adjust the range selected if you go too far. The select mode will be canceled if you move the cursor above the first selected line or *past the top of the current screen*.

The second time ⌘C is pressed the selected text is copied and placed on the clipboard.

If you are unfamiliar with the idea of a clipboard, this is just an analogy to how you might put piece of paper clipped from a magazine, letter, etc. on a clipboard, to hold it temporarily while you were getting ready to put it in its final location. The clipboard just refers to a memory buffer that holds the text you have selected while you decide where you want the final text placed. Using the clipboard, you can cut text from one source file, load another, and then paste the text into a second file. The clipboard is cleared when a file is assembled.

If text has been selected, you can press the Right arrow key and the selection will be extended to the middle of the next page. This method can be used to quickly select a large section of the listing. Do not hold the Right arrow down continuously or the keyboard buffer will fill and the selection will continue after you release the Right arrow key.

⌘D (Delete current line)

This command deletes the current line and places it in a special 'undo' buffer which is independent of the clipboard.

The ⌘R command replaces the current line with the contents of the 'undo' buffer. Therefore, to move a single line to another location, you could place the cursor on the line to be moved, and then type ⌘D to delete the line. Then move the cursor to another line, press Return, ⌘I or ⌘Tab to create an empty line, and press ⌘R to replace that line with the deleted line. Also see ⌘Delete.

⌘E (Global Exchange, also called 'Find & Replace')

Sometimes called 'Find & Replace,' this command will let you search for a group of words, and replace them with another. The ⌘E command opens a dialog box that asks for the text to change, and the new text to replace it. If you press Return alone for either of these, the command is cancelled.

If you enter the text in both fields and press Return, the file is then searched for the change text. Unlike the FIND command, it looks only for full words. That is, the text found must be bounded by non-alphanumeric characters or it will be ignored. However, if you press and hold the Open-Apple key down when you press the Return key, the exchange will operate on *all* strings found, not just on full words.

If text is found with this method, the screen is reprinted with the replacement made and the cursor is placed on the first character of the replacement. Now you must press a key to continue. Pressing Return, or most any other control character, will defeat the change and the command will look for the next occurrence of the text to change. Pressing the space bar or any other character, except A, will accept the change and the routine will continue.

You can back out of the global exchange while the cursor is on an entry by pressing Control-X. You can also press the A key, which will cause *all* occurrences to be changed.

You can tell when the routine is finished by the fact that during the exchange sequence, the line number at the top right is missing. The line number will return when there are no more matches for the change text, or when you press Control-X.

Normally, the Exchange function will find and replace all occurrences of a word throughout the source listing. This can be modified in several ways.

1. To search only the listing, and ignore words in the comments field, use the TRON command before starting the search and replace. See *Command Box Commands* for details.
2. To search just a portion of a listing, first select just the portion you wish to search as though you were going to copy it, that is, use ⌘C and the arrow keys. Then instead of copying the text with another ⌘C, use ⌘E, ⌘F, etc. to search just the highlighted text.

3. Normally, the Exchange function searches for complete words, that is, characters bounded by spaces or other non-alphanumeric characters. Suppose, however, that you wished to replace LDA #LABEL with LDA LABEL. Because this is a search string of two words, a variation is required. To search for a literal character string that may be part of a larger word, hold down the closed Apple or Option key as you press the Return key after you've input the replacement string. Thus, for our example, you could enter:

Change: LDA #L

To: LDA L

⌘F (Find text)

The ⌘F command opens a window which asks for the find text. It then finds the first occurrence of the text in the entire text file. The text can be anywhere on a line. After the first find, you can find the next occurrence by typing another ⌘F. You can edit the line and then type ⌘F to go to the next occurrence.

If there are more occurrences to be found, one or more plus (+) signs will be shown next to the line number at the top right of the screen. This starts from the line below the current line, and only indicates the number of lines remaining with occurrences, and not the total number of occurrences.

If the ⌘F command is used after text has been selected, only the selected text will be searched for the text to be found. When the search has been completed, the text is no longer selected. Thus, you can use the ⌘Y, ⌘C or ⌘X commands to search just a portion of your listing. You can also use the TRON command before starting the search to ignore comments during the search.

Control-X, and the ⌘B command and Control-S status command both cancel the Find mode, as does failure to find the text below the current line.

The ⌘W command is identical to ⌘F except that it finds only whole words bounded by non-alphanumeric characters. If you type either ⌘W or ⌘F to find the next occurrence, this mode will change accordingly.

In all cases the line containing the text is moved to the center of the screen, unless it is within the first 10 lines of the start of the source.

⌘H (Half screen)

Pressing ⌘H toggles the half or split screen mode. In this mode, the bottom ten lines are frozen in a window. A bar is shown above these lines to separate the frozen text from the scroll window. Pressing ⌘H will cancel the half screen mode and refresh the screen.

␣I (Insert line)

Pressing ␣I will insert a blank line at the cursor. Also see ␣TAB.

␣L (Locate label, marker or line number)

This command will locate the first occurrence of a label or any text in the label column. Only the characters typed are compared with the labels. Thus, the search string *LOOP* would jump to the label *LOOP2* if *LOOP* did not occur first. To find a specific label when there may be other similar labels, end the input with a single space.

If a number is entered after this command, the cursor will move to the beginning of the line number specified. This is particularly handy when editing a source file from a printed listing.

The intended use for this command is to move rapidly to a particular place in the source. You can use create your own 'markers' to enhance the capability of this command. Thus, if a line has **7* in it, you can specify **7* as the text to find for this command and it will locate it.

In all cases the line containing the text is moved to the center of the screen, unless it is within the first 10 lines of the start of the source.

␣N (End of source)

This command moves the cursor to the end of the source listing.

␣O (Open Command Box)

This command opens the Command Box, from which various non-editing commands are issued. See the section on *Command Box Commands* for details. Pressing Return alone will cancel this command.

␣Q (Quit to Main menu)

This command quits the Editor and returns to the Main Menu.

⌘R (Replace)

This command exchanges the current line with the contents of the 'undo' buffer. Therefore, pressing ⌘R a second time will cancel the effect of the first press.

Using ⌘R when the cursor is on a blank line will place the contents of the 'undo' buffer on the line and place the empty line in the 'undo' buffer.

The ⌘R command can be used to move a single line. Place the cursor at the beginning of the line to be moved and press ⌘R. Move the cursor to the desired location, press Return to insert a blank line, and press ⌘R again.

⌘R can be used by itself to easily interchange two lines. Just place the cursor on the first line, press ⌘R, move the cursor to the second line and press ⌘R again. Then move the cursor back to where the first line was and press ⌘R for the third, and final time. Also see ⌘D.

⌘T (Jump to marker line - Vertical "Tab")

This command returns to the line 'remembered' by the last Control-T command.

⌘V (Paste)

Pastes the contents of the clipboard at the line containing the cursor. Only full lines are moved. Using this command does not change the contents of the clipboard, so this command can be used to replicate a range of lines.

If the ⌘V paste command is issued when a range of text has been selected, the text on the clipboard will be inserted before the last line of selected text.

⌘W (Find word)

The ⌘W command is identical to ⌘F except that it finds only whole words bounded by non-alphanumeric characters. If you type either ⌘W or ⌘F to find the next occurrence, this mode will change accordingly. Either search mode can be cancelled by pressing Control-X.

If the ⌘W command is used after text has been selected, only the selected text will be searched for the word to be found. When the search has been completed, the text is no longer selected.

⌘X (Cut)

This command is similar to Copy (⌘C), but selected text is removed from the screen after being copied to the clipboard. The first time ⌘X is pressed, the current line is selected and is shown in inverse. Use the down arrow or Escape keys to extend the selection if desired, or press any other key to cancel the selection. Additional selected lines are shown in inverse. Use the up arrow key to adjust the range selected if you go too far. The select mode will be canceled if you move the cursor above the first selected line or past the top of the current screen.

The second time ⌘X is pressed the selected text is cut and is placed on the clipboard.

One use for this command is to use ⌘B to move to the beginning of the listing, then ⌘Y to select all of the text to the end of the listing, and then ⌘X to cut the entire listing and place it on the clipboard. Pressing any other key will cancel the select mode.

If text has been selected, you can press the Right arrow key and the selection will be extended to the middle of the next page. This method can be used to quickly select a large section of the listing. Do not hold the Right arrow down continuously or the keyboard buffer will fill and the selection will continue after you release the Right arrow key.

⌘Y (Select text)

This command selects all text to be cut or copied from the current line to the end of the listing. ⌘X or ⌘C will cut or copy the text, while pressing any other key will cancel the selection.

This technique can be used to move the entire listing to the clipboard.

⌘Z (Center screen)

This command repositions the screen so that the line the cursor is on becomes the centered line on the screen.

⌘Delete (Delete)

This command deletes the line above the cursor and places it in a special 'undo' buffer which is independent of the clipboard.

␣Escape (Exit Full Screen Editor)

This command will exit the Full Screen Editor to a line-oriented Command Mode, but only if the controlling parameter in the PARMS file has enabled this function. Unless you have a strong desire to use a Command Mode similar to that for Merlin 8, this should never be needed.

␣TAB (Insert line)

Pressing ␣TAB will insert a blank line at the cursor. Also see ␣I.

␣Down arrow (Move half-screen down)

Moves the cursor down 10 lines; that line then becomes the center line on the screen. This command has the effect of moving the current line to the top of the screen and then moving the cursor to what was the bottom line on the screen.

␣Up arrow (move half-screen up)

Moves the cursor up 10 lines; that line then becomes the center line on the screen. This command has the effect of moving the current line to the bottom of the screen and then moving the cursor to what was the 1st line on the screen.

␣Right arrow (Move one page down)

Moves the cursor down 24 lines; that line then becomes the center line on the screen.

␣Left arrow (move one page up)

Moves the cursor up 24 lines; that line then becomes the center line on the screen.

␣1 (PRTR 1 + Assemble)

This combination command issues a PRTR 1 followed by an ␣A to assemble. Thus ␣1 will activate the printer with the PRTR 1 command and then assemble and print the current listing. The default printer init string from the PARMS file, if any, is used with ␣1 and there will be no page break title unless the source file sets one with the TTL opcode. This command is provided for convenience.

⌘2 (PRTR 1 + USER + Assemble)

This combination command issues a PRTR 1 followed by USER and an ⌘A to assemble. ⌘2 will activate the printer with the PRTR 1 command, send the USER command, and then assemble and print the current listing. This command is provided for convenience. See the USER description in the Utility Programs section.

⌘3 (PRTR 3 + Assemble)

This combination command issues a PRTR 3 followed by an ⌘A to assemble. Thus ⌘3 will activate the screen with the PRTR 3 command and then assemble and print the current listing to the screen instead of the printer. This command is provided for convenience.

⌘4 (PRTR 3 + USER + Assemble)

This combination command issues a PRTR 3 followed by an USER and an ⌘A to assemble. ⌘4 will activate the screen with the PRTR 3 command, send the USER command, and then assemble and print the current listing to the screen instead of the printer. This command is provided for convenience. See the USER description in the Utility Programs section).

⌘6 (LINK for LINKER.GS only)

This command is the equivalent of typing LINK in the Command Box. ⌘6 can be used with the LINKER.GS only. This command is provided for convenience.

⌘8 (Asterisk)

Produces a line of 32 asterisks. Overstrikes existing line, if any. Undo with the Control-R command.

⌘9 (Box)

Produces an asterisk, 30 spaces, and then another asterisk. This and the ⌘8 command can be used to produce a large box for titles and other information. Overstrikes existing line, if any. Undo with the Control-R command.

⌘- (Hyphen)

Produces a line of 1 asterisk and 31 hyphens. Overstrikes existing line, if any. Undo with the Control-R command.

⌘= (Equal sign)

Produces a line of 1 asterisk and 31 equal signs. Overstrikes existing line, if any. Undo with the Control-R command.

⌘ [Closed Apple]

This command speeds up cursor movement and keyboard input. For example, pressing and holding down the down arrow key will scroll the listing. If you press and hold down the Closed Apple key while you press the up or down arrow key, the scroll rate will be increased.

MERLIN 16 COMMAND BOX

GENERAL GUIDELINES FOR THE COMMAND BOX COMMANDS

The Command Box is used for various commands not directly related to editing the current listing. To open the Command Box from the Full Screen Editor, you would press CO , and then type in the desired command. For most of the Merlin 16 Command Box operations, only the first letter of the command is required, the rest being optional. This manual will show the required command characters in UPPER case and the optional ones in lower case.

ABOUT THE COMMAND BOX DOCUMENTATION

For each of the commands available from the Merlin 16 Command Box, the documentation consists of three basic parts:

- 1) the name and syntax of the command
- 2) examples of the use of each available syntax
- 3) a description of the function of each command

When the syntax for each command is given:

PARENTHESES $()$ indicate a required value

ANGLE BRACKETS $\langle \rangle$ indicate an optional value or character

SQUARE BRACKETS $[]$ are used to enclose comments about the command

Delimited Strings (or d-strings)

Several commands allow specification of a string. The string must be delimited by a non-numeric character other than the slash or comma. Such a string is called a delimited or d-string. The usual delimiter is single or double quote marks (' or "). Delimited string examples:

'this is a delimited string'

"this is a delimited string"

@this is another d-string@

Note that the slash '/' cannot be used as a delimiter since it is the character that delimits range lists in line number-related commands.

Wild Card Characters in Delimited Strings

For all of the commands that use delimited strings (d-strings), the '^' character acts as a wild card character. For example, the d-string 'Jon^s' is equivalent to both 'Jones' and 'Jonas' d-strings.

Upper and Lower Case Control

The shift and caps lock keys work as you would expect. While editing or entering a line of text, you can also use the Control-L command, described earlier in this section.

Line Numbers in Command Box

With some commands, you must specify a line number, a range of line numbers, or a range list. A line number is just a number. A range is a pair of line numbers separated by a comma. A range list consists of several ranges separated by a slash (/).

Line Number examples:

10	LINE #	[a single line number]
10,30	RANGE	[the range of lines 10 to 30]
10,30/50,60	RANGE LIST	[ranges 10 to 30 AND 50 to 60]

If a line number in a range exceeds the number of the last line in the source, the editor automatically adjusts the specified line to the last line number. For example, if you wanted to List all the lines past 100 in a source listing, entering L100,9999 would do it.

COMMAND BOX COMMANDS

Following are the commands recognized by Merlin 16 in the Command Box of the Full Screen Editor. The Command Box is opened by pressing CO from within the Full Screen Editor. Pressing Return alone will cancel and close the Command Box. Pressing Return after entering one of the following commands will execute the command.

NOTE: UPPER case characters shown in command are required, lower case characters are optional and are listed for purposes of clarity only.

A (Add)

A [only option for this command]

The Add command places you at the end of the existing source listing, if any, and is equivalent to the CN command.

ASM (Assemble)

ASM [only option for this command]

This passes control to the assembler, which then assembles the source file.

Assembly may be terminated at any point by pressing Control-C or Escape.

During the second pass of assembly, pressing Control-D will toggle the list flag, so that the listing will either stop or resume. The next LST opcode in the source overrides this, but another Control-D can be used again.

FIX

FIX [undoes the TEXT command]

FIXS [same but removes multiple spaces except in comments
and ASCII strings]

This undoes the effect of the TEXT command. It also does a number of technical housekeeping chores. It is recommended that FIX be used on all source files from external sources that are being converted to Merlin 16 source files, after which the file should be saved.

NOTE: FIX will truncate any lines longer than 255 characters.

GET

GET (obj adrs)

GET	[put object code in Main Memory at the address specified in the source's ORG]
-----	---

GET \$4000	[put object code at location \$4000 in Main Memory]
------------	---

This command is used to move the object code, after an assembly, from its location in Auxiliary Memory to its ORG location in Main Memory. The address must be above the existing source file, if any, and it will not be allowed to overwrite DOS. You can do a NEW if you want to load it lower in memory than allowed, but you must remember to save the source first. You cannot use GET to put object code at memory locations lower than \$901, but you can go to the Monitor afterwards and use it to move the object to any desired location. However, any such move using the Monitor may destroy your source or other data necessary to the assembler's operation. Caution should be used.

The GET command does not check if a valid object code has been assembled.

This command is supplied for convenience only. The recommended method for testing a program is to save the source code first, save the object code, and then run the program from BASIC or with the G command from the Monitor.

HEX-DEC CONVERSION

128 = \$0080

\$80 = 128

If you type a positive or negative decimal number in the Command Box, the hex equivalent is returned. If you type a hex number using the \$ prefix, the decimal equivalent is returned. All commands accept hex numbers.

L (List)

List

List (line number)

List (range)

List (range list)

L	[list entire file]
L 20	[list line 20 only]
L 20,30	[list 20 through 30]
L 20,30/40,42	[list 20 through 30 and then list lines 40 through 42]

Lists the source file with line numbers. Control characters in source are shown in inverse, unless the listing is being sent to a printer or other nonstandard output device.

The listing can be aborted by Control-C or with the slash (/) key. You may stop the listing by pressing the space bar and then advance a line at a time by pressing the space bar again. By holding down the space bar, the auto-repeat feature of the Apple will result in a slower listing. Any other key will resume the normal speed. This space bar technique also works during assembly and the symbol table printout. Any other key will restart it.

MON (Monitor)

MON [only option with this command]

This exits to the Monitor. You may return to the Merlin 16 Main Menu by pressing Control-C, Control-B, or Control-Y. These commands re-establish important zero page pointers from a safe area inside Merlin 16. Thus Control-Y will give a correct entry even if you have damaged the zero page pointers while in the Monitor. DOS is not connected while using this entry to the Monitor.

You may return to the editor directly by typing 0G and pressing Return, but unlike the above commands, this uses the zero page pointers stored at \$0A-\$0F. Therefore, you must be sure that these pointers have not been altered. For normal usage, any of the three Control commands should be used to return to Merlin 16.

NOTE: When you exit to the Monitor with the MON command, the RAM-based \$D000-\$FFFF memory is enabled, and therefore, Merlin 16 and its symbol table if any. If you want to examine the ROM memory that would normally correspond to Applesoft and the F8 Monitor, you should quit Merlin 16 with the Main Menu Quit command, and enter the Monitor with Call -151. This procedure necessitates loading BASIC.SYSTEM which removes Merlin 16 from memory.

NEW

NEW [only option for this command]

Deletes the present source file in memory. Also see UNNEW.

P (Print without line numbers)

Print

Print (line number)

Print (range)

Print (range list)

P	[print entire file]
P50	[print line 50 only]
P50,100	[print lines 50 through 100]
P1,10/20,30	[print 1 through 10 and then print lines 20 through 30]

This is the same as LIST except that line numbers are not added. See PRTR for formatted printouts.

PRTR (Formatted printout)

PRTR (command)

PRTR 1	[activate printer in slot 1 with no printer init string]
PRTR 1 ""Page Title"	[printer in slot 1, no printer init string, "Page Title" is the page header]
PRTR 1 "<Control-I>80N"	[as above, add Control-I80N to initialize the printer]
PRTR 3	[send formatted listing to screen]

This command is for sending a listing to a printer with page headers and provision for page boundary skips. See the section on Configuration for details on setting up default parameters, also TTL in the Assembler section. The entire syntax of this command is:

```
PRTR # "(string)"<page header>"
```

If the page header is omitted, the header will consist of page numbers only.

The initialization string may not be omitted if a page header is to be used. If no special string is required, use a null string of two quotes (""), as in the example showing "Page Title" in which case a carriage return will be used. No output is sent to the printer until a LIST, PRINT, or ASM command is issued. See Control-O for information on inserting Control characters in the printer init string. The PRTR command only affects the next output command, and is canceled at the end of the listing.

Q (Quit)

Q [only option for this command]

Exits to Main Menu.

SYM (Symbol table)

SYM	[print symbol table from last assembly]
SYM'TR'	[print only those labels starting with 'string' specified]
SYM:	[print local labels in symbol table]
SYM:'TR'	[print local labels with 'string' specified]

The SYM command will print the symbol table from the last assembly to be printed. Care should be taken to ensure that a valid symbol table exists before this command is used.

The SYM: command inverts the *print locals flag* and then prints the symbol table with local labels first. Thus, it is not necessary to change this bit in the PARMS file to see the local label list.

If a string is specified with either SYM and SYM:, only those labels starting with the string will be printed

NOTE: If the printing is aborted and the SYM command is reissued, only a partial listing will result. This is because symbol flagging is done during the sorting of the table. However, if the printing is allowed to continue to the end, the entire symbol table will be printed the next time SYM or SYM: is used. If aborted during an alphabetical printout, the SYM command will restart where it left off.

TABS

TABS <number><, number><,...> <'tab character'>	
TABS	[clear all tabs]
TABS 10,20	[set tabs to 10 & 20]
TABS 10,20 " "	[as above, space is tab character]

This sets up to 9 tabs for the editor, and has no effect on the assembler listing. The default tab character is a space, but any may be specified. Space is the only acceptable tab character for the separation of labels, opcodes, and operands. If you don't specify the tab character, then the last one used remains. Entering TABS and a Return will set all tabs to zero.

TEXT

TEXT	[only option for this command]
------	----------------------------------

This converts all spaces in a source file to inverse spaces. The purpose of this is for use on word processing type text files so that it is not necessary to remember to zero the tabs before printing such a file. This conversion has no effect on anything except the Editor's tabulation. The command FIX undoes the effect of the TEXT command.

TROF (Truncate Off)

TROF [only option for this command]

This command returns to the default condition of the truncation flag which also happens automatically upon entry to the editor from the Main Menu or from the Assembler. All source lines when listed or printed will appear normal.

TRON (Truncate On)

TRON [only option for this command]

This command sets a flag which, during LIST or PRINT, will suppress printing of comments that follow a semicolon. It makes reading of some source files easier. Typing TRON before using CE or CF also causes the search to ignore text within comments.

UNNEW

UNNEW [only option for this command]

This command restores a text file cleared by a NEW command. Note that the first two characters of the source file cannot be restored, so some additional editing will be necessary. Also see NEW.

USER

USER
USER 1 [example for use with XREF]
USER 0: FILENAME [example for use with PRINTFILER]

This does a JSR to the routine at \$3F5. This is the location of the Applesoft ampersand vector which normally points to a RTS. USER is designed to connect the various utilities supplied with Merlin 16 and for user defined printer drivers. You must be careful that your printer driver does not use zero page addresses, with the exception of the I/O pointers and \$60-\$6F and \$90-\$9F, because this is likely to interfere with Merlin 16's heavy usage of zero page. Several supplied utilities operate through the USER command. Specifically, XREF is a USER type utility. Also see the description of USER files supplied with Merlin 8/16 as described in the Utilities section of this manual.

VAL

VAL "expression"

VAL "LABEL" [gives the address (or value) of LABEL for the last assembly
done or "unknown label" if not found.]

VAL "\$1000/2" [returns \$0800]

This will return the value of the expression as the assembler would compute it (or was used in the last assembly). All forms of label and literal expressions valid for the assembler are valid for this command.

VID (VIDeo)

VID (slot number)

VID 3 [turns on 80 column display]

This command selects an 80 column display device. To turn off the display, use Escape Ctrl-Q for the Apple IIe, IIc or IIgs video; use Escape-0 for the Videx UltraTerm. VID 3 is required to re-activate the display if you have set the PARMs file to switch to 40 columns for the PRTR command.

W (Where)

Where (line number)

W 50 [where is line 50 in memory]

W 0 [where is end of source file]

This prints in hex the location in memory of the start of the specified line. *Where 0* or *W0* will give the location of the end of source.

. [period]

. [only option for this command]

Lists starting from the beginning of the last specified range. For example, if you type *L10,100*, lines 10 to 100 will be listed. If you then use the period (.) command, the listing will start again at 10 and continue until stopped. The end of the range is not remembered.

/ (slash)

/ <line number>

/ [start to list at last line listed]

/50 [start listing at line 50]

This command continues the listing from the last line number listed, or, when a line number is specified, from that line. This listing continues to the end of the file or until it is stopped as in LIST.

MERLIN 16 EDITOR COMMAND SUMMARY

CONTROL KEY COMMANDS (line oriented)

The Control Key commands consist of cursor moves and line oriented commands.

Control-B	Moves cursor to beginning of line
Control-C	Cancel assembly and return to Editor at last line assembled
Control-D	Deletes character under the cursor
Control-F	Finds next occurrence on current line of next character typed
Control-I	Toggles insert and overstrike cursor
Control-L	Toggles lower case conversion
Control-N	Moves cursor to end of line
Control-O	Prefix key for typing optional characters
Control-R	Restores original line
Control-S	Displays status box
Control-T	Remember current line for recall by ⌘T
Control-W.....	Finds next occurrence of word in line
Control-X.....	Exits global exchange, etc. while in progress
Control-Y	Delete characters to end of line
Arrows	Moves the cursor in the specified direction
Delete	Deletes character to left of cursor
Escape	Moves cursor to beginning of next line
Return	Moves cursor down and inserts blank line
TAB	Toggles insert and overstrike cursor

OPEN-APPLE KEY COMMANDS (entire listing oriented)

The Open-Apple Key commands are global commands, which means they are generally oriented to the whole listing as opposed to just the current line (or a single character).

⌘A	Assembles the current source listing
⌘B	Moves cursor to beginning of listing
⌘C	Start text selection/Copy selected text to clipboard
⌘D	Deletes line and places it in 'undo' buffer
⌘E	Global exchange (Find & Replace)
⌘F	Finds next occurrence of text entered
⌘H	Toggles half or split screen mode
⌘I	Inserts blank line at cursor
⌘L	Finds first occurrence of label or line
⌘N	Moves cursor to end of listing
⌘O	Opens Command Box
⌘Q	Quits Editor and returns to Main Menu
⌘R	Swaps current line with 'undo' buffer
⌘T	Goes to line of last Control-T
⌘V	Pastes contents of clipboard on current line
⌘W	Finds next occurrence of whole word
⌘X	Start text selection/Cut selected text to clipboard
⌘Y	Selects text from current line to end of file
⌘Z	Current line becomes eleventh line on screen
⌘Delete	Deletes line above cursor; puts in 'undo' buffer
⌘TAB	Inserts a blank line at cursor
⌘Down	Moves cursor down 10 lines
⌘Up	Moves cursor up 10 lines
⌘1	Issues 'PRTR 1' + 'Assemble'
⌘2	Issues 'PRTR 1' + 'USER' + 'Assemble'
⌘3	Issues 'PRTR 3' + 'Assemble'
⌘4	Issues 'PRTR 3' + 'USER' + 'Assemble'
⌘6	Issues 'LINK' command to LINKER.GS
⌘8	Produces a line of 32 asterisks
⌘9	Produces 1 asterisk, 30 spaces, and 1 asterisk
⌘-	Produces a line of 1 asterisk followed by 31 hyphens
⌘=	Produces a line of 1 asterisk followed by 31 equal signs
⌘	Increases speed of cursor movement and keyboard input

COMMAND BOX COMMANDS

The Command Box commands are a series of commands not directly related to editing the listing. These commands are always preceded by an Open-Apple-O [⌘O] which opens the Command Box. After entering the desired command, press Return. UPPER case characters are required, lower case characters are optional and are listed for purposes of clarity.

Add	Moves to the end of the current listing
ASM	Assembles the current source file
FIX	Undoes the effect of the TEXT command
GET	Place object code in Main Memory at specified address
Link	Assemble and link last source file worked on.
List	Lists line or range of lines
MON	Exits Editor and enters Monitor
NEW	Clears current source file from memory
Print	Prints listing without formatting
PRTR	Prints formatted listing. See command for complete syntax
Quit	Exits to Main Menu
SYM	Prints symbol table of last assembly
TABS	Sets tabs used by the Editor
TEXT	Zeroes all tabs in text file
TROF	Turns off truncation of comments field in printout
TRON	Turns on truncation of comments field in printout
UNNEW	Restores text file deleted by the NEW command
USER	Connects various utilities and user defined printer drivers
VAL	Returns value of expression or address of label
Where	Prints memory location of specified line
. (period)	Lists from beginning of last specified range
/(slash)	Continues listing from last specified line

CUSTOMIZING MERLIN 16 COMMANDS

Almost all of the Merlin 16 Editor command keys can be changed if you have a preference for an alternate definition. See the description of the PARMS file for Merlin 16 in the Technical Information section of this manual.

GENERAL REMARKS ON MERLIN 16 EDITOR

When you move the cursor between lines, its horizontal position may vary. This is because the cursor position is based on the actual position in memory in the line, and not on the screen position. If the tabs are zeroed you will not notice this, except for the fact that the cursor is never beyond the last character in the line.

The maximum line length is actually 192 but you can only edit the first 80 characters. *Lines longer than 80 characters will be truncated if they are edited.* This can be important if you are using TXTED as mentioned in the Utilities section.

An assembly will delete the contents of the clipboard.

EDMAC, TXTED, and AUTO.EDIT

A macro program called EDMAC is also included on the Merlin 16 disk, which will automatically generate phrases like "LDA # \$" in your source file with a single key command. TXTED is a modified full screen editor that breaks lines when you press Return, and can delete Returns as well. AUTO.EDIT is an automated text editor that can speed up making the same change in a number of places in a source file where the Find or Exchange commands would not be appropriate. See the description of these utilities in the Utilities section of this manual for more information.

OOPS

Virtually any editor action can be undone. You should remember that the proper undo command is of the same 'type' as the command you want to undo. Thus, any Control key command is undone by Control-R. This includes the $\text{C}8$, $\text{C}9$, $\text{C}-$, and $\text{C}=\text{C}$ commands which are considered line oriented commands for this purpose.

The line deletion commands $\text{C}D$ and $\text{C}Delete$ are undone by creating an empty line with $\text{C}Tab$ followed by $\text{C}R$. If you forget to create the empty line, type another $\text{C}R$ and then insert the empty line to receive the undo buffer contents.

The $\text{C}R$ command undoes itself.

A Cut ($\text{C}X$) is undone by a Paste ($\text{C}V$) without moving the cursor off its line.

If you are entering a line of text in response to a prompt, such as a filename, PRTR initialization, or dialog box, you can press Control-C or Control-X to cancel the line.

ASSEMBLING A MERLIN 16 FILE

Once you have entered and edited your source listing, you will want to assemble it. Just press Open-Apple-A. This passes control to the assembler, which attempts to assemble the source file.

If you wish to have a formatted printed listing of an assembly, just use the PRTR command immediately before using the Open-Apple-A command.

Pressing Control-C terminates the assembly and returns to the Editor with the cursor on the line where assembly was interrupted.

THE ASSEMBLER

In Merlin 8/16, the Editor is used to create and edit the source listing from which the final program, or object code, will be assembled. The Assembler is that part of Merlin 8/16 which actually interprets your source code to create the final program.

The Assembler portion of Merlin 8/16 is distinct only in concept. In practice, both the Editor and Assembler are resident in the machine at all times, and thus both are available without having to be aware of which is in operation at any given time. This is in contrast to many other assemblers, in which the Editor and Assembler are completely separate programs, necessitating the switching between them by loading and running independent programs, and often requiring that you save the source file to disk before an assembly can even be done.

This section of the documentation explains the syntax of those commands, or directives, that can be used in the source listing itself, and which direct Merlin 8/16 to perform some function while assembling the object code. These are in contrast to the Editor commands which are used primarily to edit an existing line of text.

For example, in the simplest assembler possible, only commands like LDA, JSR, etc. would be recognized by the assembler. However, the first time you want to create a data table, an instruction is required by the assembler which will define one or more bytes that are a pure number value, as opposed to specific opcodes. This is addressed in virtually all assemblers by creating the assembler directive, or pseudo opcode HEX.

Thus the assembler can create a byte of data like this:

```
1 LABEL    HEX F7    ; STORES BYTE '$F7'
```

Now, suppose the data you wanted to store was an ASCII character string. With only the HEX directive, you'd have to look up all the ASCII character equivalents, and encode them in your program with individual HEX statements.

Wouldn't it be nice, though, if the assembler itself had a larger repertoire of new commands or directives that included ones for defining character strings? You bet! And Merlin 8/16 has a lot of them.

The simplest is ASC, and a typical line would look like this:

```
1 LABEL    ASC 'THIS IS A TEST' ; STORE ENTIRE CHARACTER STRING
```

When assembled, Merlin 8/16 would automatically look up the ASCII character equivalents, and store the bytes in memory at wherever that statement occurred in your program. Along with the Editor, the variety and power of assembler directives is the other biggest factor in determining the power of a given assembler. Merlin 8/16 is outstanding in this area with a wide complement of directives for every occasion.

This section of the documentation will explain the syntax to use in your source files for each directive, and document the features that are available to you in the assembler.

ABOUT THE ASSEMBLER DOCUMENTATION

The assembler documentation is broken into three main sections:

- 1) Preliminary Definitions
- 2) Assembler Syntax Conventions
- 3) Assembler Pseudo Opcode Descriptions

The last two sections are each broken down further into the following:

Assembler Syntax Conventions:

- 1) Number Format
- 2) Source Code Format
- 3) Expressions Allowed by the Assembler
- 4) Immediate Data Syntax
- 5) Addressing Modes
- 6) Sweet 16 Opcodes
- 7) Native vs. Emulation mode in Merlin 16

Assembler Pseudo Opcode Descriptions:

- 1) Assembler Directives
- 2) Formatting Pseudo Ops
- 3) String Data Pseudo Ops
- 4) Data and Storage Allocation Pseudo Ops
- 5) Miscellaneous Pseudo Ops
- 6) Conditional Pseudo Ops
- 7) Pseudo Ops for Macros
- 8) Variables

The Assembler Syntax Conventions illustrate the syntax of a line of assembly code, the proper method to specify numbers and data, how to construct assembler expressions and the proper syntax to use to specify the different addressing modes allowed by the 65xx microprocessors. This section should be understood prior to using the assembler, otherwise it will be difficult to determine the acceptable methods to construct a proper expression as the operand for a pseudo op.

The Assembler Pseudo Opcode Descriptions illustrate the functions of the many Merlin 8/16 pseudo ops, the correct syntax to use and examples of each pseudo op's use.

PRELIMINARY DEFINITIONS

The type of operand for almost all of Merlin 8/16's pseudo ops and the 65xx microprocessors can be grouped into one of four categories:

- 1) Expressions
- 2) Delimited Strings (d-strings)
- 3) Data
- 4) Filenames or Pathnames

Expressions

Expressions are defined in the Assembler Syntax Conventions section of this Chapter.

Delimited Strings

Delimited Strings are defined in the Editor section of the manual, but that definition is repeated here for continuity.

Several of the pseudo opcodes, and some of the 65xx opcodes such as LDA, allow their operand to be a string. Any such string must be delimited by a non-numeric character other than the slash (/) or comma (,). Such a string is called a "d-string" or delimited string. The usual delimiter is a single or double quote mark (" or ').

Examples:

"this is a d-string"

'this is another d-string'

@another one@

Zthis is one delimited by an upper case zZ

"A"

'A'

NOTE: Delimited strings that are used as the object of *any* 65xx opcode *must* be enclosed in single or double quotes. If not, the assembler will interpret the d-string to be a label, expression or data instead.

Take special note that some of the pseudo ops as well as the 6502 and 65C02 opcodes use the delimiter to determine the *hi-bit* condition of the resultant string. In such cases the delimiter should be restricted to the single or double quote.

Data

Data is defined as raw hexadecimal data composed of the digits 0 through 9 and the letters A through F.

Filenames (DOS 3.3 only)

Filenames are defined as the name of a DOS 3.3 file without any delimiters, e.g. no quotes surrounding the name. Source file names are suffixed with a .S while a T. is used as the prefix for Text files, USE files and PUT files. The applicable suffix or prefix *should not* be used as part of the filename when loading, saving, reading or writing.

Pathnames (ProDOS only)

Pathnames are defined as ProDOS pathnames and are restricted to the definition of pathnames as described in the ProDOS User's Manual. Pathnames as used by Merlin 8/16 do not have delimiters, e.g. no quotes surrounding the pathname. The .S suffix is used for source, USE, and PUT pathnames. This suffix *should not* be used as part of the pathname when loading or saving.

ASSEMBLER SYNTAX CONVENTIONS

SOURCE CODE FORMAT

Syntax of a Source Code Line

A line of source code typically looks like:

```
LABEL      OPCODE  OPERAND      ;COMMENT
```

and a few real examples:

```
1  START      LDA  #50           ;THIS IS A COMMENT
2  *          THIS IS A COMMENT ONLY LINE
3                                     ;TABBED BY EDITOR
```

A line containing only a comment can begin with an asterisk (*) as in line 2 above. Comment lines starting with a colon (;), however, are accepted and tabbed to the comment field as in 3 above. The assembler will accept an empty line in the source code and will treat it just as a SKP 1 instruction, except that the line number will be printed. See the section on pseudo opcodes for details.

The number of spaces separating the fields is not important, except for the editor's listing, which expects just one space.

Source Code Label Conventions

The maximum allowable LABEL length is 26 characters in Merlin 16 or 13 characters in Merlin 8, but more than 8 will produce messy assembly listings unless you change the tab settings. A label must begin with a character at least as large, in ASCII value, as the colon, and may not contain any characters less, in ASCII value, than the number zero. Note that periods (.) are not allowed in labels since the period is used to specify the logical OR in expressions.

A line may contain a label by itself. This is equivalent to equating the label to the current value of the address counter.

Source Opcode and Pseudo Opcode Conventions

The assembler examines only the first 3 characters of the OPCODE, with certain exceptions such as macro calls and the DEND opcode. For example, you can use PAGE instead of PAG. However, because of the exception, the fourth letter should not be a D. The assembler listing will not be aligned with an opcode longer than five characters unless there is no operand or you change the tab settings.

Operand and Comment Length Conventions

The maximum allowable combined OPERAND + COMMENT length is 64 characters. You will get an OPERAND TOO LONG error if you use more than this. A comment line by itself is also limited to 64 characters.

LOCAL LABELS

A local label is any label beginning with a colon (:). A local label is attached to the last global label and can be referred to by any line from that global label to the next global label. You can then use the same local label in other segments governed by other global labels. You can choose to use a meaningless type of local label such as :1, :2, etc., or you can use meaningful names such as :LOOP, :EXIT, and so on.

Example of local labels:

```

1  START  LDY #0
2         LDX #0
3  :LOOP  LDA (JUNK),Y      ;:loop is local to start
4         STA (JUNKDEST),Y
5         INY
6         CPY #100
7         BNE :LOOP      ;branch back to :LOOP in 3
8  LOOP2  LDY #0
9  :LOOP  LDA (STUFF),Y    ;:loop is now local to loop2
10        STA (STUFFDEST),Y
11        INY
12        CPY #100
13        BNE :LOOP      ;branch back to :LOOP in 9
14        RTS

```

Some restrictions on use of local labels are:

- 1) Local labels cannot be used inside macros.
- 2) You cannot label a MAC, ENT or EXT with a local label and you cannot EQUate a local label.
- 3) The first label in a program cannot be a local label.

Local Labels, Global Labels and Variables

There are three distinct types of labels used by the assembler. Each of these are identified and treated differently by Merlin.

Global Labels	: labels not starting with "]" or ":"
Local labels	: labels beginning with ":"
Variables	: labels beginning with "]"

Note that local labels do not save space in the symbol table, while variables do. Local labels can be used for forward and backward branching, while variables cannot. Good programming practice dictates the use of local labels as branch points, variables for passing data, etc.

VARIABLES

Labels beginning with a right bracket (]) are regarded as variables. They can be redefined as often as you wish. The designed purpose of variables is for use in macros, but they are not confined to that use.

Forward reference to a variable is impossible, that is, with correct results, but the assembler will assign some value to it. Therefore, a variable should be defined before it is used.

It is possible to use variables for backwards branching, using the same label at numerous places in the source. This simplifies label naming for large programs and uses much less space than the equivalent once-used labels.

For example:

```

1          LDY #0
2 ]JLOOP  LDA TABLE,Y
3          BEQ NOGOOD
4          JSR DOIT
5          INY
6          BNE ]JLOOP      ;BRANCH TO LINE 2
7 NOGOOD  LDX #-1
8 ]JLOOP  INX
9          STA DATA,X
10         LDA TBL2,X
11        BNE ]JLOOP      ;BRANCH TO LINE 8

```

NUMBER FORMAT

The assembler accepts decimal, hexadecimal, and binary numerical data. Hex numbers must be preceded by the dollar sign (\$) and binary numbers by the per cent sign (%), thus the following four numbers are all equivalent:

Dec	Hex	Binary	Binary
100	\$64	%1100100	%01100100

as indicated by the last binary number, leading zeros are ignored.

Immediate Data vs. Addresses

In order to instruct the assembler to interpret a number as immediate data as opposed to an address, the number should be prefixed with a pound sign (#). The # here stands for number or data. For example:

```
LDA #100      LDA #$64      LDA #%1100100
```

These three instructions will all load the accumulator with the number 100, decimal.

A number not preceded by # is interpreted as an address. Therefore:

```
LDA 1000      LDA $3E8      LDA %1111101000
```

are equivalent ways of loading the accumulator with the byte that resides in memory location \$3E8.

Use of Decimal, Hexadecimal or Binary Numbers

We recommend that you use the number format that is appropriate for clarity. For example, the data table:

```
DA    $1
DA    $A
DA    $64
DA    $3E8
DA    $2710
```

is a good deal more mysterious than its decimal equivalent:

```
DA    1
DA    10
DA    100
DA    1000
DA    10000
```

Similarly,

```
ORA #80
```

is less informative than

```
ORA #10000000
```

which sets the hi-bit of the number in the accumulator.

EXPRESSIONS ALLOWED BY THE ASSEMBLER

Primitive Expressions

Expressions are built up from "primitive expressions" by use of arithmetic and logical operations. The primitive expressions are:

1. A label.
2. A number (either decimal, \$hex, or %binary).
3. Any ASCII character preceded or enclosed by quotes or single quotes.
4. The asterisk character (*) which stands for the current address.

All number formats accept 16-bit data and leading zeros are never required. In case 3, the "value" of the primitive expression is just the ASCII value of the character. The hi-bit will be on if a quote (") is used and the value greater than \$7F. The hi-bit will be off if an apostrophe (') is used and the value less than \$80.

Arithmetic and Logical Operations in Expressions

The assembler supports the four arithmetic operations: +, -, / (integer division), and * (multiplication). It also supports the three logical operations: ! (Exclusive OR), . (OR), and & (AND).

Building Expressions

Expressions are built using the primitive expressions defined above, either with or without arithmetic and/or logical operations. This means that expressions can take the form of primitives or primitives operated on by other primitives using the arithmetic and logical operators.

Some examples of legal expressions are:

#01	(primitive expression = 1)
#\$20	(primitive expression = 32 dec)
LABEL	(primitive consisting of a label)
#"A"	(primitive consisting of letter "A")
*	(primitive = current value of PC)

The following are examples of more complex expressions

LABEL1-LABEL2	(LABEL1 minus LABEL2)
2*LABEL+\$231	(2 times LABEL plus hex 231)
1234+%10111	(1234 plus binary 10111)
"K"-"A"+1	(ASCII "K" minus ASCII "A" plus 1)
"0"!LABEL	(ASCII "0" EOR LABEL)
LABEL&\$7F	(LABEL AND hex 7F)
*-2	(current address minus 2)
LABEL.%1000000	(LABEL OR binary 1000000)

By clever use of the simple arithmetic and logical operators in carefully designed expressions, you can create other functions that may not be immediately obvious.

For example, for a conditional assembly, you might want to see if one label had a value greater than that of another. Although Merlin 8/16 doesn't have a specific < or > function, you can still do the equivalent test using the division operator:

```
DO LABEL1/LABEL2      ; 0 IF LABEL1 < LABEL2 = DON'T DO
                       ; 1 IF LABEL1 >= LABEL2 = DO
DO LABEL1/5            ; DO IF LABEL1 >= 5
DO LABEL1-1/-1        ; DO IF LABEL1 = 0
DO 5-1/LABEL1         ; DO IF LABEL1 < 5
DO LABEL1/6           ; DO IF LABEL1 > 5
```

As another example, use of the AND operator (&) makes it simple to define a control character:

```
CHK LDA CHAR          ; INPUT CHARACTER
    CMP #$9F&"A"      ; CONTROL-A
    BNE NEXT
```

\$9F AND \$C1 ("A") = \$81 ("Control-A")

Checking for equality is a simple subtraction:

```
DO LABEL1-5-1/-1      ; DO ONLY IF LABEL1 = 5
```

Parentheses and Precedence in Expressions

Parentheses are not normally allowed in expressions. They are not used to modify the precedence of expression evaluation. All arithmetic and logical operations are evaluated left to right. Thus, $2+3*5$ would assemble as 25 and not 17.

Parentheses are used to retrieve a value from the memory location specified by the value of the expression within the parentheses, much like indirect addressing. This use is restricted to certain pseudo ops, however.

For example:

```
DO ($300)
```

will instruct the assembler to generate code if the value of memory location \$300 is non-zero at the time of assembly.

Example of Use of Assembler Expressions

The ability of the assembler to evaluate expressions such as LAB2-LAB1-1 is very useful for the following type of code:

```
COMPARE    LDX    #EODATA-DATA-1
LOOP       CMP    DATA,X
           BEQ    FOUND      ;found
           DEX
           BPL    LOOP
           JMP    REJECT     ;not found
DATA       HEX    CACFC5D9
EODATA     EQU    *
```

With this type of code, you can add or delete some of the DATA and the value which is loaded into the X index for the comparison loop will be automatically adjusted.

IMMEDIATE DATA SYNTAX

For those opcodes such as LDA, CMP, etc., which accept immediate data, i.e. numbers as opposed to addresses, the immediate mode is signed by preceding the expression with #. An example is LDX #3. When programming the 65802 or 65816, the interpretation of an immediate data expression may depend on the assembler status of the M and X bits at that point in the assembly. In general:

#expression	produces the low byte of the expression in 8 bit mode
#expression	produces the low word of the expression in 16 bit mode
#<expression	produces the low byte of the expression in 8 bit mode
#<expression	produces the low word of the expression in 16 bit mode
#>expression	produces the high byte of the expression in 8 bit mode
#>expression	produces the high word of the expression in 16 bit mode
#^expression	produces the bank byte of the expression always.

ADDRESSING MODES

Because of the different processors in use in the Apple IIe, IIc, and IIgs machines, Merlin 8/16 not only supports all the addressing modes of the 65xx microprocessors, but the different assemblers are customized to the systems they are most likely to be used on.

Merlin 8

When either version of Merlin 8 is started, the assembler assumes a 6502 microprocessor state for assemblies. If you are assembling code for a 65C02 processor, *you must use the assembler XC opcode once* at the beginning of the listing anywhere before an actual 65C02 opcode is used. The requirement for the XC enable directive is a safety feature, so that you don't inadvertently use a 65C02 opcode like STZ or INC in a program designed for a 6502 machine. Because the assembler would otherwise generate no errors with the extended opcodes, a program that unwittingly used an improper code would be very hard to debug.

If you are using Merlin 8 to write a program for the 65802, *you must use two XC pseudo-ops* at the beginning of the source listing to enable the 65802 instructions. Because the 65802 does not support more than 64K of addressable memory, Merlin 8 does not feature any long addressing modes, although macros to duplicate these functions can be written.

Merlin 16

Because Merlin 16 is most likely to be used on an Apple IIgs, the startup default for the assembler logic is to have 65816 opcodes enabled, thus avoiding the need for using the XC directives. If you will be using Merlin 16 to assemble code for Apple IIe compatible computers, you may want to consider changing the PARMs file for Merlin 16 to require the use of the XC directives as a safety device to prevent accidental use of 65816 opcodes in programs written for the Apple IIe. Even Apple Computer is not above these hazards, as evidenced by the use of a 65C02 instruction in a version of ProDOS that was intended to have been compatible with 6502 machines.

In regards to the M and X bits of the 65816, the assembler logic of Merlin 16 starts out assuming the processor will be in the Emulation Mode, since this is the power-up state of the microprocessor. Remember, this is an assumption about *your* source code, and has nothing to do with the actual program code of Merlin 16 itself. Like the XC default, you can change the PARMs file to start a source listing in any setting of the M and X bits you prefer.

When programming for the 65816, Merlin 16 uses the following definitions for address expressions:

LDA	expression	use one (low) byte of the expression
LDA	<expression	use one (low) byte of the expression
LDA	>expression	use two bytes (low word) of the expression
PEA	^expression	use two bytes (high word) of the expression
LDA	!expression	use three bytes (complete address) of the expression

this last example is equivalent in Merlin 16 to:

LDAL	expression	use three bytes (complete address) of the expression
------	------------	--

Merlin 16 allows the use of the 4th character in the opcodes ADC, AND, CMP, EOR, LDA, ORA, SBC and STA to force the long addressing modes, as shown by the previous example.

In addition, the instructions JML and JSL are always assembled in the long form, and JSR and JMP are always assembled in the short or 64K address form.

Special Forced Non-Zero Page Addressing

There is no difference in syntax for zero page and absolute modes. The assembler automatically uses zero page mode when appropriate. Merlin 8/16 provides the ability to *force* non-zero page addressing. The way to do this is to add anything except D in Merlin 8, or L in Merlin 16, to the end of the opcode. Example:

```
LDA $10  assembles as zero page (2 bytes: A5 10)
```

while,

```
LDA: $10  assembles as non-zero page (3 bytes: AD 10 00)
```

Also, in the indexed indirect modes, only a zero page expression is allowed, and the assembler will give an error message if the "expr" does not evaluate to a zero page address.

NOTE: The Accumulator Mode does not require an operand (the letter "A"). For example, to do an LSR of the accumulator, you can use:

```
1 LABEL LSR          ; LOGICAL SHIFT RIGHT
```

Some assemblers perversely require you to put an "A" in the operand for this mode.

Merlin 8/16 will decide the legality of the addressing mode for any given opcode.

Sweet 16 Opcodes (Merlin 8 only)

The Merlin 8 assembler accepts all Sweet 16 opcodes with the standard mnemonics. The usual Sweet 16 registers R0 to R15 do not have to be EQUated and the R is optional. For the SET opcode, either a space or a comma may be used between the register and the data part of the operands; that is, SET R3,LABEL is equivalent to SET R3 LABEL. It should be noted that the NUL opcode is assembled as a one-byte opcode the same as HEX 0D, and not a two byte skip, since this would be interpreted by ROM Sweet 16. This is intentional, and is done for internal reasons.

NOTE: The Sweet 16 opcodes will not be recognized by Merlin 8 unless the SW pseudo opcode has been previously assembled. This pseudo op will enable assembly of Sweet 16.

65C02 and 65802 Opcodes

The Merlin 8 and Merlin 16 assemblers accept all the 6502, 65C02 and 65802 opcodes with standard mnemonics. It also accepts BLT (Branch if Less Than) and BGE (Branch if Greater or Equal) as pseudonyms for BCC and BCS, respectively. The XC pseudo opcode activates these features. This opcode is discussed in the following section on pseudo ops.

You will have problems if you do not use the standard 65C02 opcodes as specified by GTE, NCR, and Rockwell. In creating the IIC Reference Manual, Apple apparently did not check with the manufacturers regarding the final set of opcodes. Thus, Apple refers to two *non-standard* opcodes, INA and DEA.

To increment and decrement the Accumulator, use INC and DEC with no operand, as is consistent with other Accumulator directed commands such as LSR, ASL, etc.

Branch on Bit Set (BBS) and Branch on Bit Reset (BBR) are also *non-standard* Rockwell opcodes and are not supported by the NCR and GTE chips.

Merlin 16 supports the alternate opcodes:

Standard Opcode:	Alternate Opcode:
TCS	TAS
TSC	TSA
XBA	SWA
TCD	TAD
TDC	TDA
BCC	BLT
BCS	BGE