

Familiarization Exercise

1 Introduction	1
2 Design	2
3 Edit	3
4 Compile/Assemble	4
4.1 Executing the DEFT Pascal Compiler	4
4.2 Executing the 6809 Macro Assembler	5
5 Link	7
6 Execute/Debug	8



1 Introduction

In order to illustrate the use of the **DEFT Pascal Workbench**, a sample program has been included on the diskette. This program is made up primarily of a **PASCAL** program which is contained in the files *FORMAT/PAS* and *FORMAT2/PAS*. An assembler module *FORMATSP/ASM* contains a pre-initialized lookup table that is used by the Pascal program. The assembler module has already been assembled into an object file (*FORMATSP/OBJ*), however, if you also have **DEFT Bench**, then you can also perform the section on assembling a program.

2 Design

This step has already been performed for you. The purpose of the program is to read an ASCII file, which can be created by **DEFT Edit**, and to produce a professional looking document. The input file for this program contains text and text processing commands which control how the resulting document is to look. Text processing commands are recognized by having a period (.) as the first character in a line. The document that will be produced as a result of this exercise, contains a *Detailed Functional Description* of what the program is to do.

The program is broken down into the following major procedures:

- *Initialize* initializes all variables and prompts for file names required.
- *ReadNextLine* reads the next line of input and determines whether it is a command or text. If it is a command, it determines which command that it is.
- *NextSymbol* parses an input command for each parameter of that command.
- *FillOutput* and *NoFillOutput* create normal output text from an input text line.
- One procedure per command will be used to process each command type.

3 Edit

This phase has also been performed. As mentioned before, the files *FORMAT/PAS* and *FORMAT2/PAS* contain the Pascal program. The file *FORMATSP/ASM* contains the assembly language support for the program. With **DEFT Edit** or your own ASCII file text editor, you can edit these files to see what they look like. We recommend that you don't make any changes to the program until after you have made a backup and have executed the final program at least once successfully.

4 Compile/Assemble

We are now ready to compile the Pascal program and assemble the assembler support code. This section assumes that you are using a two disk drive system with the **DEFT Pascal Workbench** diskette in drive 0 and your work diskette in drive 1.

If you have only a single drive system, then you will have to copy the following files onto your work diskette (see the section on *Single Drive Operation*):

FORMAT/PAS
FORMAT2/PAS
FORMATSP/ASM
FORMATSP/OBJ
FORMAT/TXT

Before starting make sure that you have performed the steps described under *Getting Started* to protect the machine language programs from BASIC.

4.1 Executing the DEFT Pascal Compiler

The command `LOADM "PASCAL":EXEC` will load the **DEFT Pascal Compiler** from disk drive 0 and begin execution. You will see the **DEFT Pascal Compiler** screen with all of its prompts. If you have only a single disk drive, then remove the **DEFT Pascal Workbench** diskette from the drive and insert your work diskette. Each prompt and its possible replies are described below:

- **SOURCE** requires the name of the source file which is to be compiled. The default extension is PAS. Your response for this sample program will be `FORMAT`, `FORMAT:0`, `FORMAT/PAS` or `FORMAT/PAS:0` all of which are equivalent.
- **OBJECT** requires the name of the object file that is to be created by the compiler. This can be either on tape or disk or the name can be omitted entirely if you do not wish to create an object file. The default extension is OBJ. Your response for this sample program will be `FORMAT:1` or `FORMAT/OBJ:1` both of which are equivalent. If you have a single drive system, your response will be `FORMAT`, `FORMAT/OBJ` or `FORMAT/OBJ:0`.
- **LIST** requires the name of the list file which is to be created by the compiler. This can be tape, disk, screen or printer or the name can be omitted entirely if you do not wish to create a list file. The default extension is LIST. Your response for this sample program

will be :-2 if you have a printer or nothing if you don't.

- **DEBUG?** asks you whether you wish to have debug information included in the resulting object file. You can answer this either with *N*, *n* or anything else. Anything other than *N* or *n* (for No) is taken to be *Y* (for Yes).

The debug information will make your program significantly bigger but will allow you to symbolically debug your resulting program if you answer the **DEFT Linker's** *debug?* question with a *Y*. If you specify *Y* to **DEFT Pascal's** *debug?* question and *N* to the **DEFT Linker's** debug question, then the debug information will still be in the final binary image even though the **DEFT Debugger** module is not present.

If you want to try out the debugger, then you can answer this question *Y*, otherwise answer it *N*.

- **DIRECTIVE** requires any **DEFT Pascal** directive that you would like to include before any source lines are read. The section *Compiler Controls* describes all the possible compiler controls that you could enter here. Your response for this sample program will be *T<your name>* which will cause *<your name>* to be printed at the top of each page of the program listing.

After you answer the **DIRECTIVE** prompt, the program will begin executing. The compiler requires that the file **PASCALIB/EXT** be present on disk drive 0 at this point. When the compiler is finished executing, control will return to **BASIC** and you will get the **OK** prompt.

This execution of the **DEFT Pascal Compiler** will read both the **FORMAT/PAS** and **FORMAT2/PAS** source files and create the **FORMAT/OBJ** object file. The **FORMAT2/PAS** file will be read because of a compiler directive at the end of the **FORMAT/PAS** source file.

4.2 Executing the 6809 Macro Assembler

If you want to try out the **DEFT Macro/6809** assembler then you can also assemble **FORMATSP/ASM** into the **FORMATSP/OBJ** file. If you don't, then go to the next section.

First put the **DEFT Pascal Workbench** diskette in disk drive 0 and enter the command **LOADM "ASSEMBLE":EXEC** to load **DEFT Macro/6809** and begin its execution. If you have a single drive

system, put your work diskette into disk drive 0.

You will see the assembler's screen appear along with its first prompt. Each prompt and its possible replies are described below:

- **TITLE:** requires the string of characters that you want to see at the top of each page of your assembly listing. You do not have to enter a title but for this sample program you can enter your name.
- **SOURCE FILE:** requires the name of the source file which is to be assembled. The default extension is ASM. Your response for this sample program will be *FORMATSP*, *FORMATSP:0*, *FORMATSP/ASM* or *FORMATSP/ASM:0* all of which are equivalent.
- **OBJECT FILE:** requires the name of the object file that is to be created by the assembler. This can be either on tape or disk or the name can be omitted entirely if you do not wish to create an object file. The default extension is OBJ. Your response for this sample program will be *FORMATSP:1* or *FORMATSP/OBJ:1* both of which are equivalent. If you have a single disk drive system, your response will be *FORMATSP*, *FORMATSP:0*, *FORMATSP/OBJ* or *FORMATSP/OBJ:0*.
- **LIST FILE:** requires the name of the list file which is to be created by the assembler. This can be tape, disk, screen or printer or the name can be omitted entirely if you do not wish to create a list file. The default extension is LST. Your response for this sample program will be *:2* if you have a printer or nothing if you don't.

After you answer the *LIST FILE:* prompt, the assembler will begin its first pass. During this first pass only the disk will appear to be doing anything. For this sample program, the first pass should last only a few seconds. The assembler will begin printing on its second pass through the source code. During this second pass **DEFT Macro/6809** will read the *FORMATSP/ASM* source file and produce the *FORMATSP/OBJ* object file and a listing on your printer.

5 Link

Once you have created the necessary object files with the compiler and assembler, you are ready to link them together into your final binary image. Make sure that you have the **DEFT Pascal Workbench** diskette in disk drive 0 and then enter the command *LOADM "LINKER":EXEC* to load **DEFT Linker** and begin its execution. If you have a single drive system, put your work diskette in disk drive 0. The *Operation* section in the **DEFT Linker** documentation describes how to operate the Linker. For your sample program, the responses required will be:

- **ORIGIN** - no response, this will invoke the default origin.
- **LIST FILE:** - :*2* if you have a printer, otherwise nothing.
- **BINARY FILE:** - *FORMAT:1* or *FORMAT/BIN:1* both of which are equivalent. If you have a single drive system, enter *FORMAT, FORMAT:0, FORMAT/BIN* or *FORMAT/BIN:0* all of which are equivalent.
- **PASCAL? (Y)** - Y.
- **DEBUGGER? (Y)** - Y if you want to try out **DEFT Debugger**, otherwise N.
- **OBJ NAMES FILE:** - no response, this is because you do not have a text file that contains the file names of all the object files to be linked.
- **OBJECT FILE:** - *FORMAT:1* or *FORMAT/OBJ:1* both of which are equivalent. If you have a single drive system, enter *FORMAT, FORMAT:0, FORMAT/OBJ* or *FORMAT/OBJ:0* all of which are equivalent.
- **OBJECT FILE:** - *FORMATSP:1* or *FORMATSP/OBJ:1* both of which are equivalent. If you have a single drive system, enter *FORMATSP, FORMATSP:0, FORMATSP/OBJ* or *FORMATSP/OBJ:0* all of which are equivalent.
- **OBJECT FILE:** - no response to indicate that you have entered all the object file names that you wish to link.

The Linker will then begin operation and produce both the final binary image in the file *FORMAT/BIN* and a listing on your printer.

6 Execute/Debug

The command *LOADM "FORMAT:1":EXEC* (*LOADM "FORMAT":EXEC* on a single drive system) will load the sample program and begin its execution. If you specified *Y* to the *DEBUGGER?* prompt from **DEFT Linker** then you will see the **DEFT Debugger** screen. The **DEFT Debugger** documentation provides a complete description of how to operate the debugger. If you did not specify *Y* or if you give **DEFT Debugger** the *GO* command, then you will see the *FORMAT* screen with its first prompt. You should answer the prompts as follows:

1. **INPUT FILE:** - *FORMAT*, *FORMAT:0*, *FORMAT/TXT* or *FORMAT/TXT:0* all of which are equivalent.
2. **OUTPUT FILE:** - *-:2* if you have a printer. If not, put the output on disk by entering *FORMAT:1* or *FORMAT/LST:1* both of which are equivalent. If you have a single disk system use *FORMAT*, *FORMAT:0*, *FORMAT/LST* or *FORMAT/LST:0* all of which are equivalent.

Once you answer the last prompt the program will begin executing and produce a document showing you how to use the program.