DISCLAIMER

PLEASE READ BEFORE OPENING SEALED DISCETTE ENVELOPE

The installation of HIER is not for beginners. It is assumed that the purchaser has a good knowledge of assembly language programming. I cannot be held responsible for the installation of HIER of any data lost using HIER. Liability is strictly limited to the original license cost.

Read this manula thoroughly and decide if you can carry out the necessary software changes to install HIEr. If not, return the HIER discette unopened, and your money will be refunded (less shipping and damage charges).

ONCE THE DISCETTE ENVEKOPE HAS BEEN OPENED NO RETURNS WILL BE ACCEPTED

HIERARCHICAL FILE SYSTEM

Ray Goff
134 Robin Hill Rd.
Chelmsford MA 01824
Tel. 617-256-5216

INTRODUCTION

One of the delights of FLEX is the friendly interface it offers to the computer. It was, however, originally designed in the heyday of the eight-inch floppy disc. In those days drives were expensive and it seemed unthinkable that anyone would own more that four.

Today Winchester drives are no more expensive than eight-inch drives were then. Although a single Winchester drive can hold more files than four eight-inch drives it is simply not possible to come up with enough unique names to fill it, and if one did, the time taken to search the directory would be prohibitive.

It was from these frustrations and the desire to continue to use FLEX that HIER was born.

Having a hierarchical file directory system means that a single drive can be segmented into a number of smaller directories. For example if you are an avid C programmer you can keep a C directory which can in turn hold the directories of all your different C programmes. Another directory can be set up for assembly language programmes, again with a directory for each program.

The advantages are that it is now possible to have a large number of directories each holding a few programmes. The limitation of file names being unique is then only true in a given directory - FLEX is not aware of other directories on the disc.

Each directory looks to FLEX like a file with the extension .DIR.

If it sounds too good to be true, be advised that there are some disadvantages. It is only reasonable to make small changes to FLEX which means that some overhead is required for the system to function correctly.

FLEX is only aware of one directory per disc and the command line interpeter is not capable of decoding a directory path. Also, since the sub-directories are transparent to FLEX, additional disc accesses are required to ensure that FLEX is always working in the correct directory.

All the additional disc reads required to keep the hierarchical directory system working are handled by HIER.CMD when it is activated.

Do not be too dismayed the utilities that come with HIER make up for just about all the limitations.

SYNTAX

The syntax that I have adopted for HIER consists of inserting the path to the directory containing the file in square brackets between the optional file number and the drive name. The original FLEX directory I have called the HOME directory, and a period is used to delimit each directory in the path. You are not required to specify a path for files in the current directory. For example to list a file called backup.c in the backup directory on the work drive simply type:

```
+++list [home.c.backup]backup.c
```

The only difference between this and the usual synax for list is the directory description contained within the square brackets. This assumes that the HOME directory (the FLEX directory at track zero sector five) contains a file C.DIR which in turn contains a file called BACKUP.DIR which holds the file BACKUP.C. The number of directories on a disc is limited only by the storage.

If you simply enter:

```
+++list [.backup]backup.c
```

it will be assumed that the backup directory is below the current directory, that is, from the example above you are currently in the C directory.

LIMITATIONS

The only program that I am aware does not function correctly is QCHECK. it is unable to correctly list the files in the print queue if they are not in the current directory. Instead int generates an error and exits. Therefore if you enter files into the print queu and then change directory you cannot use the facilities of QCHECK unless you return to the original directory. The files will still print correctly; they just cannot be listed.

This problem can be overcome simply by copying the files to the system disc before printing them - assuming that you are less likely to want to change directories on the system disc.

A second limitation is that in order to change directories, or access files in another directory the HIER utilities must be able to WRITE to the system information record on track zero sector three - which means that the disc cannot be write-protected during these operations.

Because HIER installs itself at the top of user memory it is very important that programs take note of the FLEX memory limit so that they do not overwrite HIER.

HIER automatically installs itselk at the top of user memory whenever it is invoked. In oder to ensure HIER does not take up more memory than it needs, it should not be invoked from an EXEC file. This is because the storage EXEC uses at the top of user memory is not released if HIER has been installed on top.

To avoid this problem I use the following text in my STARTUP.TXT file

```
HOME 0:HIER:EXEC START
```

where START contains al the programs I execute at startup.

ADVANTAGES

Just in case the advantages of HIER have not become fully obvious, let me list just a few for you:

1). The limitation of four disc drives is removed; each disc can be sub-divided into any number of directories.

2). With smaller directories, the time taken to open a file is greatly reduced.

3). Backing up your work is made easier because only the files in the directory you have been working on need to be backed up.

4). The limitation on unique file names is only true in a directory and not across the entire disc.

ADAPTATION

INTRODUCTION

HIER is a collection of files which when properly installed add the power of a hierarchical directory system to FLEX.

HIER has been designed for easy installation, but to allow the installer to estimate if the amount of work required is more than he / she is capable of, an installation program is provided on a seperate disc. Running this program will give an assessment of the amount of work required. Once you have run the install program and determined that the installation exceeds your skill level then HIER can be returned for a refund IF THE ENVELOPE CONTAINING THE HIER DISC HAS NOT BEEN OPENED.

To better understand how HIER interfaces to FLEX, and therefore how to install it, it is first necessary to understand a little more of the inner workings of FLEX.

The FLEX file management system, FMS, is responsible for managing the files on a disc. To do this it sub-divides each disc into two sections - system and user.

The system section, which occupies track zero, is further divided into three sub-sections.

1). The bootsrap, which normally occupies the first sector.

2). The system information record, sector three, which maintains a record of the number and position of free sectors on the disc.

3). The directory, which extends from sector five to the end of the track and holds a record of all the files on the disc along with the sectors allocated to them.

The user section occupies the remaining sectors on the disc and holds the user files.

Whenever a file is opened, FMS searches the directory for the file. To do this is first executes an open directory call to point to the start of the directory, followed by get information calls to read every entry in the directory. Each file-name is then compared against the requested name until it is found - or in the case of a new file, the end of directory is detected.

When the hierarchical directory software is running HIER intercepts FMS open directory calls to substitute the current directory pointer (which is stored in the system information record) for track zero sector five - where FMS usually expects to find the directory.

HIER utilities which change directories such as SETDIR manipulate the directory pointer in the system information record.

Since an empty directory looks to FLEX like a file of unused sectors (ones filled with hex zero), creating a new directory is simply a matter of creating a file filled with hex zeros.

Initially all sub-directories are made four sectors long, enough for 40 files, but this is not a limitation because FLEX will extend the directory provided spare sectors exist on the disc.

The installation of HIER consists of two parts

1). Identifying the address within FLEX of the FMS open directory routine so that a jump to HIER can be inserted.

2). Identifing the addresses within FLEX that contain 0005 - the FLEX start of directory sector - so that HIER can substitute the track and sector of the current directory.

The installation program, INSTALL, scans through FLEX to identify the addresses within FLEX that HIER needs to operate successfully.

If any addresses in your version of FLEX differ form the addresses in the version of HIER.CMD provided as part of this software package, the new addresses must be edited into HIER.TXT. This new HIER.TXT must then be assembled using the FLEX assembler ASMB to generate a working version of HIER.CMD.

HIER.CMD must then be executed each time FLEX is booted up. It moves itself into the top of user memory, protects itself by adjusting the FLEX top of memory pointer, and patches the changes into FLEX so that it gets called whenever an FMS open dirctory call is executed.

INSTALLATION

The installation program, INSTALL, on the disc marked installation will assess the amount of installation work required as outlined above. Before running it take a look at the listing of HIER.TXT in the appendix. You will notice that it has three sections of code which potentially need modifing depending on the outcome of running the install program.

After your version of FLEX has been booted, remove the disc marked INSTALLATION from its sleeve and insert it into drive zero.

Type:

```
+++0.install
```

to run the installation program. First is should print:

```
install v1.1
```

followed by:

```
processing section one
```

if there are no changes to section one it will print:

```
no changes necessary to section one
```

otherwise it will print:

```
FIND5x    EQU   $Dxxxx

edit HIER.TXT to change this value
```

it will then print

```
     processing section two
```

here it will print out one of three messages

```
     no changes necessary to section two
```

or

```
     READSEC   EQU  $Dxxx
     OPNDIR    equ  $Dxxx

     edit these changes into HIER.TXT
```

or

```
     Install was unable to to identify the required addresses
     refer to the section DIGGING DEEPER for more
     instructions.
```

finally, provided it can identify the addresses in section two it will print:

```
     processin section three
```

and either:

```
     no changes necessary in section three.
```

or

```
     further invsetigation of open directory routine required

     xx xx xx xx xx
```

where x is a hexadecimal number.

If INSTALL indicates that no further changes are required to HIER.TXT proceed to the section on testing HIER.

EDITING HIER

Should INSTALL identify any changes to HIER.TXT, it will include as much information as possible to assist in editing HIER.TXT

IF AND ONLY IF YOU FEEL COMFORTABLE WITH CHANGING HIER.TXT OPEN THE SEALED HIER DISCETTE.

Remove this discette from its sleeve and place it in drive 1, put your system discette in drive 0. Listing the direcory of drive 1 should show the following files:

BACKUP.CMD	CHGDIR.CMD	COPY.CMD
DELDIR.CMD	DIR.CMD	DISCCHK.CMD
HIER.TXT	HIER.CMD	HOME.CMD
LIST.CMD	MAKEDIR.CMD	MOVE.CMD
RUN.CMD	RUN.LOW	SETDIR.CMD
TEST.DIR	TREE.CMD	UNDER.CMD
WHERE.CMD		

The output from INSTALL along with the listing of HIER.TXT, which can be found in the Appendix, should enable you to make any changes that are necessary.

Using your editor make the changes suggested by INSTALL and proceed to the next section.

ASSEMBLING HIER

HIER.TXT should be assembled as follows:

```
+++ASMB 1.HIER 1.HIER.CMD +LSY
```

if all is well no errors will be reported so HIER can be tested.

TESTING HIER

If you arrived here without having to edit HIER.TXT now is the time to open the sealed discette, it should contain the files listed above.

FIRST MAKE A COPY OF YOUR SYSTEM DISCETTE FOR USE DURING TESTING.

With the new copy of of your system disc in drive 0, copy all the .CMD files from drive one on to it. Then type:

```
+++0.HOME 0
```

This will set the current directory pointer to home for future use. Once this has been done WRITE PROTECT both discs, then type

```
+++0.HIER
```

to invoke HIER. At this point the familiar +++ prompt should be on the terminal. If not HIER did not install correctly. So re-boot the system and check your editing.

If you feel that HIER.TXT is correct and HIER will still not install correctly, change the line:

```
DEBUG   SET 0
```
to
```
DEBUG   SET 1
```

in HIER.TXT and reassemble the file. This will cause HIER.CMD to print debugging messages as it proceeds through the installation phases.

If the FLEX prompt is on the terminal, try listing the directory of drive 1. If HIER is working correctly, the following files will appear in the directory:

```
SUCCES.TXT
NEW.DIR
```

Try listing success - it should tell you that you have sucessfully installed HIER.

If you used the new DIR to get the directory listing you should notice that the directory is TEST.

If that effort was successful, remove the write protect from drive 1

Using BUILD make a small file and make sure that is can be read back from the disc.

Now it is time to try moving to a new directory. Enter:

```
+++CHGDIR >1.NEW
```

and then obtain a directory of drive 1. This time you should see the following files:

```
SUCCESS.TXT
ANOTHER.DIR
```

List SUCCESS again - this time it should tell you that you have successfully moed to a new directory.

If all is works, try experimenting with the new commands to ensure that they all work.

CONGRATULATIONS! YOU HAVE NOW ADDED THE POWER OF A HIERARCHICAL DIRECTORY SYSTEM TO THE FRIENDLIEST OPERATING SYSTEM AVAILABLE

DIGGING DEEPER

You are unfortunate enough to have to investigate deeper into FLEX. To do this you will need a monitor program capable of examining memory - while the ability to set breakpoints will increase your chances of success.

The amount of digging around will depend on the error message from INSTALL.

SECTION ONE ERRORS

Section one of INSTALL searches through FLEX looking for occurances of the track and sector of the home directory - 0005.

Two errors are possible from SECTION ONE either INSTALL found too many occurances of 0005, or it did not find enough.

In either case it will advise which parts of HIER.TXT need changing and should not present a problem.

SECTION TWO ERRORS

Section two of INSTALL searches through FLEX to find the jump table to the FMS routines. It then validates the jumps to READ SECTOR and OPEN DIRECTORY are the same as the values in HIER.TXT. The worst error that can occur is that INSTALL was totally unable to identify the jump table - in which case it prints:

```
install was unable to identify the required addresses
read the section on DIGGING DEEPER for further advice.
```

To satisfy the equates in HIER.TXT it is necessary to find the jump table inside FLEX and enter the correct addresses for the two routines.

SECTION THREE ERRORS

Section three of the INSTALL program inspect the code at the start of the open directory code inside FLEX. If this is not the same as is

expected in HIER.TXT it will have printed the following:

```
     further investigation of the open directory routine required

     xx xx xx xx xx
```

where each xx is a hexadecimal number.

The first xx printed is the opcode which the jump to HIER will overlay it is necessary to know the length of this opcode in order to be able to adjust the return address in HIER.TXT. More detail is contained in the listing of HIER.TXT

It is difficult to give advice on how to identify the necessary code because everybody has a different approach - but it is certainly not an easy job.

Technical Systems Consultants have not changed FLEX for a number of years, so if you do see an error it is quite likely that you have an older version of FLEX. you might consider an upgrade as a simpler way of solving the problem.

If you feel that the problem is beyond your capabilities AND YOU HAVE NOT OPENED THE SEALED DISC, HIER can be returned for a refund.