CHAPTER 16. VIEWING SOURCE SCREENS

The source code about viewing is scattered in UTILITY.BLK, screen 7, EXTEND86.BLK screen 12, and KERNEL.BLK screens 66 and 76.

Forth with its hosts of definitions can be looked upon from two opposing points of view. For the Forth advocates, it is called modularity because definitions can be individually executed or compiled to build higher level definitions. For others it is called fragmentation because functions are scattered in hundreds of small bits and pieces. To decipher a colon definition, you have to know the exact functions of every word in this definition. It is not an easy job to find them because they seldom are grouped in one place.

F83 has more than 1000 words in it. How can the user find any particular word in this mess?

The designers of F83 provide us with a viewing facility which allows us to locate the source code of all the words in the dictionary and display the block of texts that contains the source of the word we look for. To accommodate this facility, the format of Forth definitions in the dictionary is changed from the traditional form of name field, link field, code field and parameter field to that as shown in Fig. 16.1.

16.1. THE VIEW FIELD

The view field is used to store information about where the source of the definition is located. It is two bytes long and divided into two sub fields: the lower 12 bits contain the block number of the source code and the upper four bits contain the file number of a DOS file containing the block. Let's first examine some of the low level words associated with the view field:

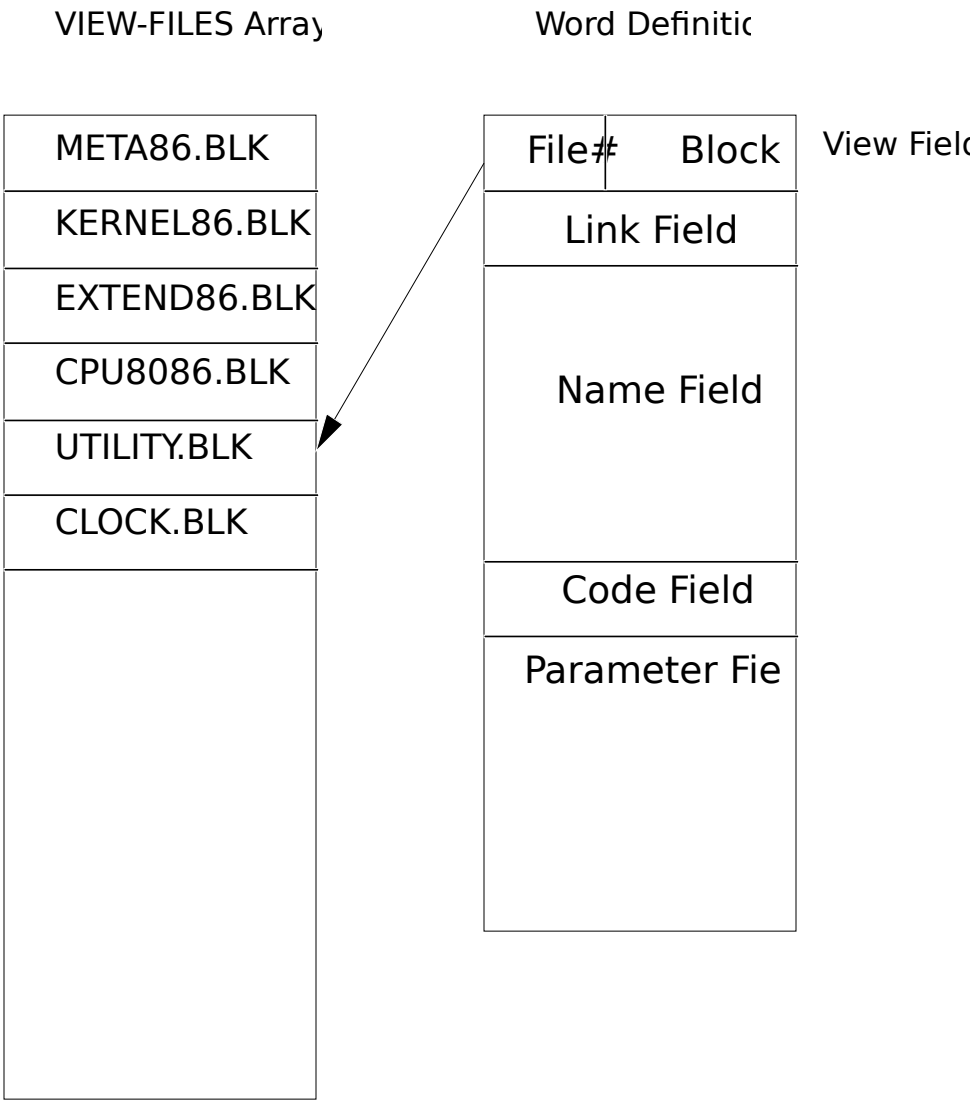
```
: VIEW#             ( --- addr )     Return the view field address in the current FCB.
      FILE @                         Address of the current FCB.
      40 +                           Offset by 40 bytes to the view field.
      ;

: ,VIEW             ( --- )          Compile the view field in a new definition.
      VIEW# @                        Get the view file number of the current file.
      4096 *                         Shift it to the upper 4 bit file number subfield.
      BLK @ +                        Add the block number of the screen under compilation.
      ;

: "CREATE           ( --- )          Create the header of a new definition.
      ALIGN ,VIEW                    Compile the view field first. The rest of the definition was
                                     discussed in the chapter on the vocabulary structure.

: >VIEW             ( cfa --- vfa )  Go from code field to the view field.
      >LINK                          Go to link field first.
      2-                             View field is two bytes ahead of the link field.
      ;
```

Figure 16.1 The view field and the view files.



```
: VIEW>             ( vfa --- cfa )  Go from view field to code field.
      2+                             To link field.
      LINK>                          From link field to code field.
      ;
```

16.2. THE VIEW FILES

The view field in a definition contains the view file number. From this number the viewing command will have to find the file and open it for viewing. The view file numbers for different source files are assigned and pertinent information is stored in an array called VIEW-FILES. In the file control block (FCB) in the definition of each source file, bytes 40 and 41 are used to store the view file number. This provides the file number to store into the view field when the definition was first compiled into the dictionary.

```
CREATE VIEW-FILES               An array where the view files are arranged in sequential
                                order so that the viewing command can find and open the
                                source file of a word definition.
      32 ALLOT                  There is enough space to define 16 files for viewing.

: VIEWS             ( n --- )   Assign the view file number on the stack to a file and store the
                                FCB address in the corresponding entry in the VIEW-FILES array.
      [ DOS ]
      ?DEFINE                   Search the dictionary for the word name following VIEWS.
                                It must be a file name and the fcb address is returned.
      2DUP 40 + !               Store number n into the view# field in the FCB of the file.
      BODY>                     Get the execution address of the file definition.
      SWAP 2*                   The address offset into the VIEW-FILES array.
      VIEW-FILES + !            Store the fcb address into the VIEW-FILES array.
      ;
```

Now, view files can be assigned view file numbers and the VIEW-FILES array be filled:

```
1  VIEWS  KERNEL86.BLK
2  VIEWS  EXTEND86.BLK
3  VIEWS  CPU8086.BLK
4  VIEWS  UTILITY.BLK
```

These are the source files in the F83 system. All the definitions loaded from these files can be viewed. If you have some application programs in another file, you will have to assign a view file number to it using the VIEWS command as above. After that, you can load in your application and can view the definitions loaded in from your file.

16.3. THE VIEWING COMMAND

Assuming that all the source code files are on the default disk drive, it is very easy to display a screen which contains the word definition you want to examine. The command is as following:

```
VIEW  <name>
```

where <name> is the name of the definition you want to review. The command VIEW will locate the word <name> in the dictionary and find out the file and the screen number of the source code of this word. It will then open that file and read the screen from the file and display the screen on the CRT terminal.

```
: @VIEW      ( cfa --- scr# file# )   From the code field address of a definition, find its view
                                      field and return the screen number and the file number stored
                                      in the view field.
        >VIEW                         Get the view field address.
        @ DUP 4095 AND                Mast off the top 4 bits in the view field and leave only the
                                      screen number.
        DUP 0=                        If the screen number is 0,
        ABORT" entered at terminal"   abort because the word is not loaded from a file.
        SWAP 4096 / 15 AND            Extract the view file number from the top 4 bits.
        ;

: VIEW       ( --- )                  Allow the user to see the source screen of the following
                                      word. If the VIEW# is zero, then the current file is used.
                                      Otherwise, the associated field is opened and viewed.
        '                             Find the cfa of the following word.
        @VIEW                         Get the screen number and the view file number.
        ?DUP IF                       If the file field is zero, use the current file.  If not zero,
                2* VIEW-FILES + @     Find the cfa of the view file in array VIEW-FILES.
                ." is in "            Print the file name first.
                2DUP >BODY .FILE
                ."  screen " .        And also the screen number.
                EXECUTE               Make the file our current file.
                OPEN-FILE             Open it for reading.
        ELSE                          The definition is in the current file.
                ."                    may be in the current file: "
                FILE?                 Print the file name,
                ." screen " DUP >     and the screen number.
        THEN
        LIST                          Show the screen.
        ;
```

View file number 0 is reserved for the current file if it has not been assigned a view file number. Up to 15 files can be stored in the VIEW-FILES array for viewing. If you have to use the second drive to store some of the files, the drive number in their FCB must be assigned accordingly.