

Firstly, you've got to remember that ZEAP is only about 2½K long and that is pretty short as assemblers come. ZEAP was designed to run in a NASCOM with the minimum (8K) expansion possible, and because of this a few short cuts were taken. The short cuts all concern what are known as 'assembler directives' which are instructions to the assembler (not program instructions to the Z80) to do certain things with certain bytes of the program. For instance a 'defined byte directive' (DEFB), now this directive instructs the assembler to put the byte you have defined into the program. So DEFB £1A means put 1A into the program (the £ sign means the byte is expressed in hexadecimal). The assembler will assemble it like this; OC50 is the address by the way 0010 is the line number:

```
OC50 1A          0010          DEFB £1A
```

Now for the short cut bit - suppose you have a string of defined bytes, followed by an instruction, the assembler would assemble them like this:-

```
OC50 1A          0010          DEFB £1A, £2C, £44, £5F
OC54 3E1F        0020          LD  A, £1F
```

Notice that ZEAP only displayed the first defined byte, but the addresses worked out right. This is because ZEAP assembled them, it just does not display them. Notice also the second instruction, there is no space between the 3E and the 1F. Remember, if you were typing that into a NASCOM, you would have to put a space between the two.

The same point about defined bytes applies to defined words (DEFW) and defined messages (DEFM) thus;

```
OC50 1A          0010          DEFB £1A, £2C, £44, £5F
OC54 350C        0020          DEFW £0C35, £0035, £0124
OC5A 54          0030          DEFM /THATS THE LOT/
OC67 3E1F        0040          LD  A, £1F
0004             0050          DEFS 4
OC6D 00          0060          NOP
```

Notice that with a defined space (DEFS) ZEAP simply put 0004 in the address field, and advanced the address counter by four.

So the basic message is, when loading programs assembled on ZEAP, watch the address counter.

Of course, lets not create a wrong impression, these short cuts in the display only occur on the monitor screen or on a printer, if you were using ZEAP yourself, these deficiencies need not cause problems, because you would ultimately assemble the program to memory or tape, in which case these bytes would be correctly assembled even though they are not displayed.