DOS 3.3: Disk layout

```
Revised:        4/8/91
Security:       Everyone
```

DOS 3.3: Disk layout

==

DOS reserves the first three tracks (0,1,2) of every disk for the boot image of itself, while track 17 ($11) is reserved for the directory and VTOC. Tracks 1 and 2 can be freed by modifying the bit map in the VTOC; however, the disk won't be able to boot properly.

Expanded the directory by changing the link bytes in bytes 1 and 2 of the last sector of the directory.

The number of sectors required for a DOS file can be calculated:

$$Sectors = length / 256 + length / 256 / 122$$

The physical number of sectors required to hold the data is recorded in the first part of the expression. It's 2 bytes higher for a program and 4 bytes higher for a binary file. The second part contains the track/sector information, even when there is no data. Thus, the minimum number of sectors for a file is two.

Random access files, as are all files, are maintained through the track/ sector list at the beginning of the file. DOS:

  * Locates the start of the record
  * Determines which sector that byte would be in, relative to the start of
    the file
  * Determines with the track/sector list in which sector the record is
    located
  * Calculates the offset into the sector
  * Writes data.

DOS allocates enough sectors in the track/sector list to access the required sector; however, DOS dosen't allocate any sectors when there hasn't been any data written to the records of that sector. Example:

```
        10 D$ = chr$(4)
        20 print D$;"open test, L500"
        30 print D$;"write test, R65"
```

```
40 print "hello"
50 print D$;"close test"
```

This program creates;

```
65 records * 500 bytes = 32500 bytes = 126 sectors

126 sectors = 2 track/sector list sectors
```

for a grand total of:

```
1 sector of data
2 sectors of track/sector list
```

No data sectors are allocated for records 0 through 64.

Apple Tech Notes

Tech Info Library Article Number:579