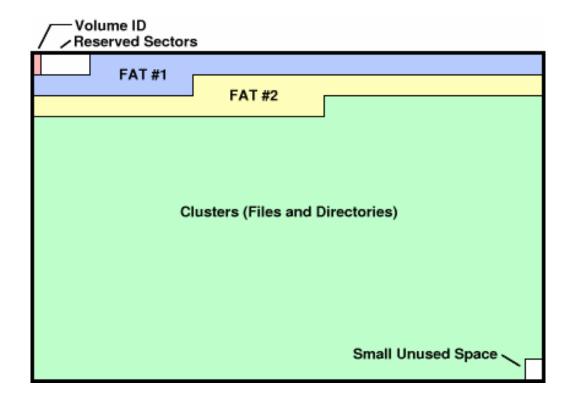
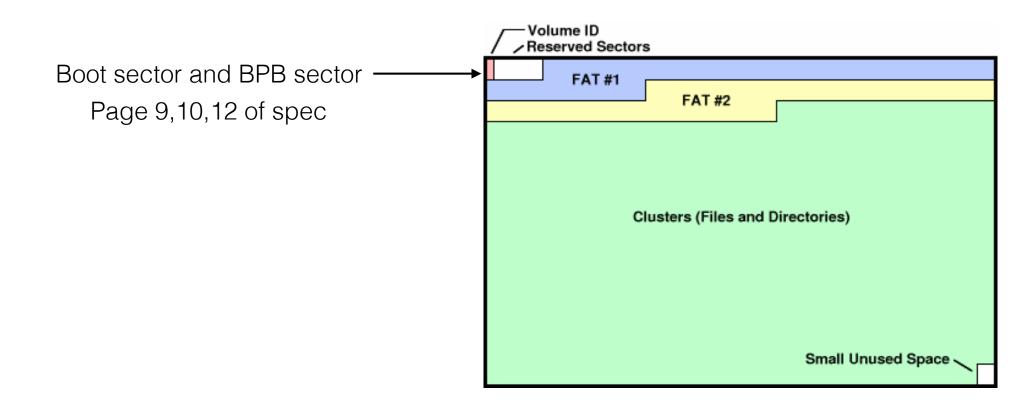
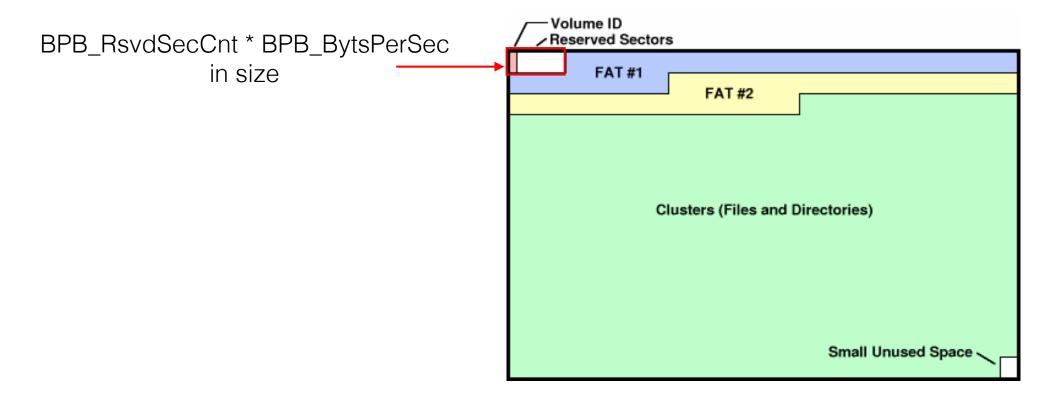
FAT32 Project

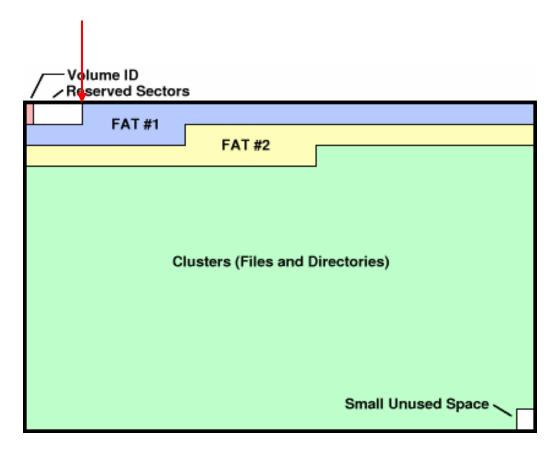


```
char BS_OEMName[8];
int16_t BPB_BytsPerSec;
int8_t BPB_SecPerClus;
int16_t BPB_RsvdSecCnt;
int8_t BPB_NumFATs;
int16_t BPB_RootEntCnt;
char BS_VolLab[11];
int32_t BPB_FATSz32;
int32 t BPB RootClus;
int32_t RootDirSectors = 0;
int32_t FirstDataSector = 0;
int32_t FirstSectorofCluster = 0;
```

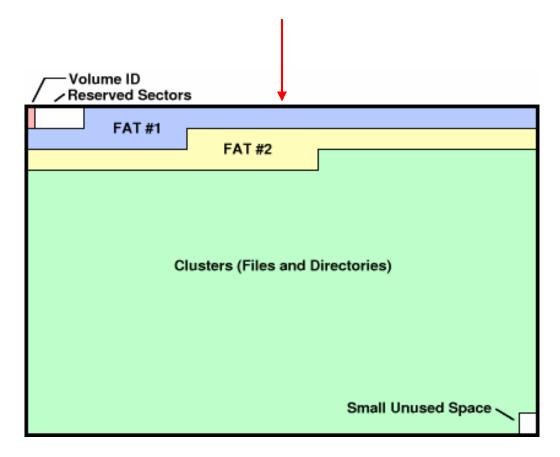




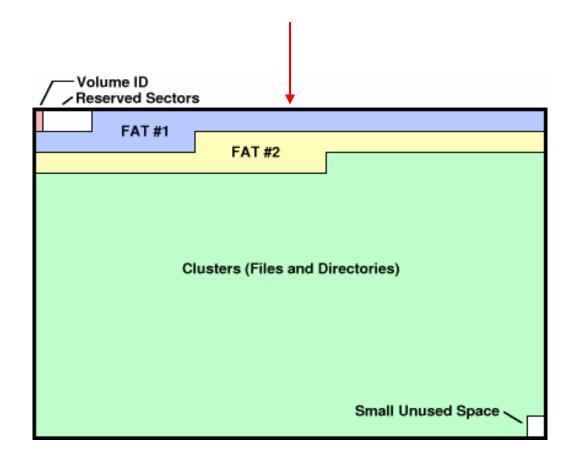
FAT #1 starts at address BPB_RsvdSecCnt * BPB_BytsPerSec



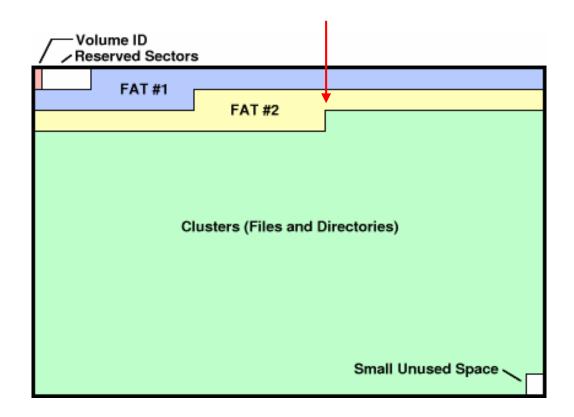
Each FAT has 1 32-bit word for every cluster. Each entry is the logical block of the next block in the file.



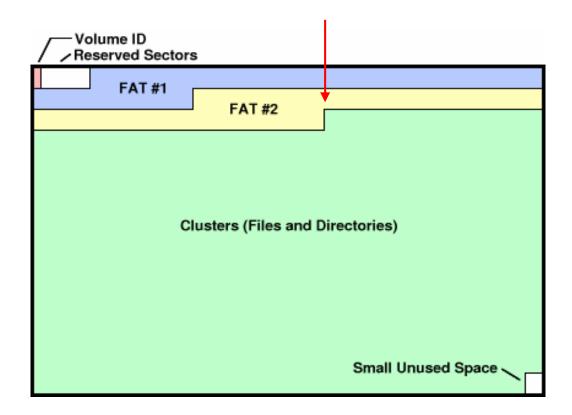
Total FAT size is BPB_NumFATs * BPB_FATSz32 * BPB_BytsPerSec



Clusters start at address (BPB_NumFATs * BPB_FATSz32 * BPB_BytsPerSec) + (BPB_RsvdSecCnt * BPB_BytsPerSec)

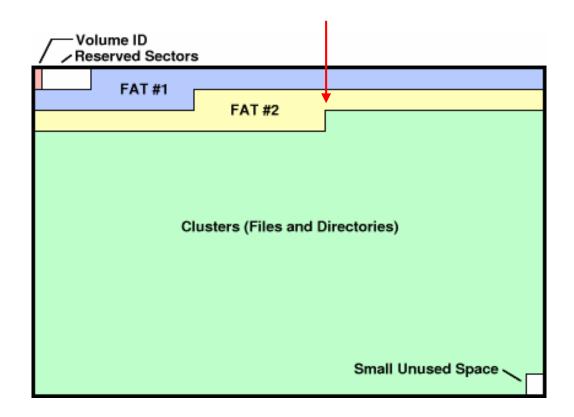


Clusters are each (BPB_SecPerClus * BPB_BytsPerSec) in bytes



Root Directory is at the first cluster.

Address: (BPB_NumFATs * BPB_FATSz32 * BPB_BytsPerSec) +(BPB_RsvdSecCnt * BPB_BytsPerSec)



Root Directory contains 16 32-byte records

Short Filename	Attrib	Cluster High	Cluster Low	Size

Field	Microsoft's Name	Offset	Size
Short Filename	DIR_Name	0x00	11 Bytes
Attrib Byte	DIR_Attr	0x0B	8 Bits
First Cluster High	DIR_FstClusHI	0x14	16 Bits
First Cluster Low	DIR_FstClusLO	0x1A	16 Bits
File Size	DIR_FileSize	0x1C	32 Bits

First character of the filename:

0x00 - Filename never used.

0xe5 - The filename has been used, but the file has been deleted.

0x05 - The first character of the filename is actually 0xe5.

0x2e - The entry is for a directory, not a normal file. If the second byte is also 0x2e, the cluster field contains the cluster number of this directory's parent directory. If the parent directory is the root directory (which is statically allocated and doesn't have a cluster number), cluster number 0x0000 is specified here.

File Attributes:

- 0x01 Indicates that the file is read only.
- 0x02 Indicates a hidden file. Such files can be displayed if it is really required.
- 0x04 Indicates a system file. These are hidden as well.
- 0x08 Indicates a special entry containing the disk's volume label, instead of describing a file. This kind of entry appears only in the root directory.
- 0x10 The entry describes a subdirectory.
- 0x20 This is the archive flag. This can be set and cleared by the programmer or user, but is always set when the file is modified. It is used by backup programs.
- 0x40 Not used; must be set to 0.
- 0x80 Not used; must be set to 0.

Only show is attribute is 0x01, 0x10, or 0x20

Each record can be represented by:

First directory entry

```
TXT .d..o>yJ....o>.....
IXI .4.CZJZJ...CZJ0....
                                               DEADBEEFTXT .uUZzJzJ..UZzJ).1...
                                               A.._...T.r....a.s.h.e.s......
                                                   TRA"...czJzJ...czJZ....
                                               .RASHE~1LZK....czJzJ...czJY.....
                                                   TXT .d..o>yJ....o>.....
                                               AF.o.l.d.e...r.A....
46 4F 4C 44 45 52 41 20 20 20 20 10 00 68 8D 44 73 4A 73 4A 00 00 8D 44 73 4A D3 17 00 00 00 00
                                               FOLDERA
                                                    ..h.DsJsJ...DsJ.....
41 2E 00 54 00 72 00 61 00 73 00 0F 00 25 68 00 65 00 73 00 00 00 FF FF FF FF 00 00 FF FF FF FF
                                               A..T.r.a.s...%h.e.s.....
54 52 41 53 48 45 7E 31 20 20 20 12 00 1B 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 59 19 00 00 00 00
                                               TRASHE~1 ....czJzJ...czJY.....
E5 31 20 20 20 20 20 20 54 45 4D 22 00 19 12 46 73 4A 79 4A 00 00 12 46 73 4A 50 19 00 10 00
                                                   TEM"...FsJyJ...FsJP.....
                                               Af.a.t.s.p...De.c..p.d.f.....
41 66 00 61 00 74 00 73 00 70 00 0F 00 44 65 00 63 00 2E 00 70 00 64 00 66 00 00 00 00 00 FF FF
46 41 54 53 50 45 43 20 50 44 46 20 00 64 38 8F 6F 3E 7A 4A 00 00 38 8F 6F 3E 0C 15 8A 4E 05
                                               FATSPEC PDF .d8.o>zJ..8.o>...N..
41 2E 00 66 00 73 00 65 00 76 00 0F 00 DA 65 00 6E 00 74 00 73 00 64 00 00 00 00 00 FF FF FF FF
                                               A..f.s.e.v....e.n.t.s.d......
46 53 45 56 45 4E 7E 31 20 20 20 12 00 20 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 62 19 00 00 00 00
                                               FSEVEN~1 ...czJzJ...czJb.....
2E 20 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E 03 00 00 00 00 00
                                                     ..dy.o>o>..y.o>.....
..dy.o>o>..y.o>.....
E5 30 20 20 20 20 20 20 20 20 20 20 00 00 D8 8E 6F 3E 57 45 00 00 D8 8E 6F 3E A6 10 00 02
                                                     ....o>WE....o>....
                                                .0
....o>WE....o>.....
                                                . 1
. 2
                                                     ....o>WE....o>.....
...G......
....o>WE....o>.....
                                                . 3
....o>WE....o>....
....o>WE....o>.....
                                                .5
....o>WE....o>.....
..dy.o>o>..y.o>.....
.dy.o>o>..y.o>....
2E 20 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E 04 00 00
2E 2E 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E C0
41 67 00 72 00 65 00 65 00 6E 00 0F 00 CA 31 00 00 00 FF FF FF FF FF FF FF FF 00 00
                                               Ag.r.e.e.n...1.....
47 52 45 45 4E 31 20 20 20 20 20 10 00 00 8A 8D 6F 3E 6F 3E 00 00 8A 8D 6F 3E 06 00 00
                                                     ....0>0>....0>....
```

Logical Block Address of this file

```
BAR
                                                          TXT .d..o>yJ....o>.....
4E 55 4D 20 20 20 20 20 54 58 54 20 18 34 8D 63 7A 4A 7A 4A 00 00 8D 63 7A 4A <mark>30 1C</mark> 00 0A 00 00
                                                     NUM
                                                          TXT .4.czJzJ...czJ0.....
44 45 41 44 42 45 45 46 54 58 54 20 18 75 55 5A 7A 4A 7A 4A 00 00 55 5A 7A 4A 29 1D 31 10 00 00 41 2E 00 5F 00 2E 00 54 00 72 00 0F 00 7F 61 00 73 00 68 00 65 00 73 00 00 00 00 00 FF FF FF FF
                                                     DEADBEEFTXT .uUZzJzJ..UZzJ).1...
                                                     A.._...T.r....a.s.h.e.s......
7E 31 20 20 20 20 20 20 54 52 41 22 00 1B 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 5A 19 00 10 00 00
                                                          TRA"...czJzJ...czJZ.....
E5 52 41 53 48 45 7E 31 4C 5A 4B 12 00 1B 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 59 19 00 00 00 00
                                                     .RASHE~1LZK....czJzJ...czJY.....
TXT .d..o>yJ....o>.....
                                                     AF.o.l.d.e....r.A.....
46 4F 4C 44 45 52 41 20 20 20 20 10 00 68 8D 44 73 4A 73 4A 00 00 8D 44 73 4A D3 17 00 00 00 00
                                                     FOLDERA
                                                          ..h.DsJsJ...DsJ.....
41 2E 00 54 00 72 00 61 00 73 00 0F 00 25 68 00 65 00 73 00 00 00 FF FF FF FF 00 00 FF FF FF FF
                                                     A..T.r.a.s...%h.e.s.....
54 52 41 53 48 45 7E 31 20 20 20 12 00 1B 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 59 19 00 00 00 00
                                                     TRASHE~1 ....czJzJ...czJY.....
E5 31 20 20 20 20 20 20 54 45 4D 22 00 19 12 46 73 4A 79 4A 00 00 12 46 73 4A 50 19 00 10 00 00
                                                         TEM"...FsJyJ...FsJP.....
41 66 00 61 00 74 00 73 00 70 00 0F 00 44 65 00 63 00 2E 00 70 00 64 00 66 00 00 00 00 00 FF FF
                                                     Af.a.t.s.p...De.c...p.d.f......
46 41 54 53 50 45 43 20 50 44 46 20 00 64 38 8F 6F 3E 7A 4A 00 00 38 8F 6F 3E 0C 15 8A 4E 05 00
                                                     FATSPEC PDF .d8.o>zJ..8.o>...N..
41 2E 00 66 00 73 00 65 00 76 00 0F 00 DA 65 00 6E 00 74 00 73 00 64 00 00 00 00 00 FF FF FF FF
                                                     A..f.s.e.v....e.n.t.s.d......
46 53 45 56 45 4E 7E 31 20 20 20 12 00 20 87 63 7A 4A 7A 4A 00 00 87 63 7A 4A 62 19 00 00 00 00
                                                     FSEVEN~1 ...czJzJ...czJb.....
2E 20 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E 03 00 00 00 00 00
                                                           ..dy.o>o>..y.o>.....
2E 2E 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E C0 17 00 00 00 00
                                                           ..dy.o>o>..y.o>.....
.f.0....
....o>WE....o>.....
                                                     .0
..J.....
E5 31 20 20 20 20 20 20 20 20 20 20 00 00 D8 8E 6F 3E 57 45 00 00 D8 8E 6F 3E A7 10 00 02
                                                            ....o>WE....o>.....
                                                     . 1
. 2
                                                            ....o>WE....o>.....
                                                     .f.3......G.....o>WE....o>.....
E5 33 20 20 20 20 20 20 20 20 20 20 00 00 D8 8E 6F 3E 57 45 00 00 D8 8E 6F 3E A9 10 00
....o>WE....o>.....
                                                     . 4
..<.....
.5
                                                            ....o>WE....o>.....
...o>WE...o>.....dy.o>o>.y.o>....dy.o>o>.y.o>....
. 6
2E 20 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E 04 00 00 00 00 00
2E 2E 20 20 20 20 20 20 20 20 20 20 10 00 64 79 8D 6F 3E 6F 3E 00 00 79 8D 6F 3E C0 17 00
   Ag.r.e.e.n...1.....
47 52 45 45 4E 31 20 20 20 20 20 10 00 00 8A 8D 6F 3E 00 00 8A 8D 6F 3E 06 00 00 00
                                                           ....0>0>....0>.....
```

- How do we get the root directory information?
- fseek to the root directory address.
- Loop from i = 0 to i < 16
 - fread 32 bytes into your directory entry array

```
/*
    *Function : LBAToOffset
    *Parameters : The current sector number that points to a block of data
    *Returns : The value of the address for that block of data
    *Description : Finds the starting address of a block of data given the sector number
    *corresponding to that data block.
    */
    int LBAToOffset(int32_t sector){
        return (( sector - 2 ) * BPB_BytesPerSec) + (BPB_BytesPerSec * BPB_RsvdSecCnt) + (BPB_NumFATs * BPB_FATSz32 * BPB_BytesPerSec);
}
```

```
/*
  Name: NextLB
Purpose: Given a logical block address, look up into the first FAT and
  return the logical block address of the block in the file. If
  there is no further blocks then return -1
  */
int16_t NextLB( uint32_t sector )
{
  uint32_t FATAddress = ( BPB_BytsPerSec * BPB_RsvdSecCnt ) + ( sector * 4 );
  int16_t val;
  fseek( fp, FATAddress, SEEK_SET );
  fread( &val, 2, 1, fp );
  return val;
}
```

- Files to help debug on the fat32.img
- num.txt 5 512-byte blocks of consecutive numbers. 512 0's followed by 512 1's ... etc
- deadbeef.txt 4145 bytes file of DEADBEEF repeating
- foo.txt "1 2 3 4"
- bar.txt "5 6 7 8"