MFS_INIT - mini-FSD Initialisation

Purpose

Inform the mini-FSD that it should prepare itself for use.

Calling Sequence

```
int far pascal MFS_INIT(pBootData, pucResDrives, pulVectorIPL, pBPB,
pMiniFSD,
                        pDumpAddr)

void far * pBootData;
char far * pucResDrives;
long far * pulVectorIPL;
void far * pBPB;
unsigned long far * pMiniFSD;
unsigned long far * pDumpAddr;
```

Where

pBootData is a pointer to the data passed from the black box to the mini-FSD (null if not passed).

pucResDrives is a pointer to a byte which may be filled in by the mini-FSD with the number of drive letters (beginning with 'C') to skip over before assigning drive letters to local fixed disk drivers (ignored if not remote IPL). The system will attach the reserved drives to the mini-FSD through a call to *FS_ATTACH* just after the call to *FS_INIT*.

pulVectorIPL is a pointer to a double word which may be filled in by the mini-FSD with a pointer to a data structure which will be available to installable device drivers through the standard device helper function *GetDosVar* (variable number 12). The first eight bytes of the structure MUST be a signature which would allow unique identification of the data by cooperating device drivers (for example, IBMPCNET).

pBPB is a pointer to the BPB data structure (see OS2LDR interface).

pMiniFSD is a pointer to a double word which is filled in by the mini-FSD with data to be passed on to the FSD.

pDumpAddr is a pointer to a double word which is filled in by the mini-FSD with the address of an alternative stand-alone dump procedure.

Returns

If no error is detected, a zero error code is returned. If an error is detected, a non-zero error code is returned.

Remarks

The mini-FSD should fill in the data pointed to by *pMiniFSD* with any 32-bit value it wishes to pass on to the FSD (see *FS_INIT*). OS/2 makes no assumptions about the type of data passed. Typically, this will be a pointer to important data structures within the mini-FSD which the FSD needs to know about.

OS/2 will not free the segment containing BootData. It should be freed by the mini-FSD if appropriate.

The DumpProcedure is a routine provided by the mini-FSD which replaces the diskette-based OS/2 stand-alone dump procedure. This routine is given control after the OS/2 kernel receives a stand-alone dump request. The OS/2 kernel places the machine in a stable, real mode state in which most interrupt vectors contain their original power-up value. If this address is left at zero, the OS/2 kernel will attempt to initiate a storage dump to diskette, if a diskette drive exists. The provided routine must handle the dumping of storage to an acceptable media.