FSH_DEVIOCTL

Purpose

This function sends an IOCTL request to a device driver.

Calling Sequence

```
int far pascal FSH_DEVIOCTL(flag, hDev, sfn, cat, func, pParm, cbParm,
pData,
                            cbData)

unsigned short flag;
unsigned long hDev;
unsigned short sfn;
unsigned short cat;
unsigned short func;
char far * pParm;
unsigned short cbParm;
char far * pData;
unsigned short cbData;
```

Where

flag indicates whether the FSD initiated the call or not.

IOflag == 0x0000	indicates that the FSD is just passing user pointers on to the helper.
IOflag == 0x0001	indicates that the FSD initiated the DevIOCtl call as opposed to passing a DevIOCtl that it had received.
All other bits are reserved.	

hDev is the device handle obtained from VPB

sfn is the system file number from open instance that caused the *FSH_DEVIOCTL* call.

This field should be passed unchanged from the sfi_selfsfn field. If no open instance corresponds to this call, this field should be set to 0xFFFF.

cat is the category of IOCTL to perform.

func is the function within the category of IOCTL.

pParm is the long address to the parameter area.

cbParm is the length of the parameter area.

pData is the long address to the data area.

cbData is the length of the data area.

Returns

If no error is detected, a zero error code is returned. If an error is detected, one of the following error codes is returned:

- ERROR_INVALID_FUNCTION – indicates the function supplied is incompatible with the category and device handle supplied.
- ERROR_INVALID_CATEGORY – indicates the category supplied is incompatible with the function and device handle supplied.
- Device driver error code

Remarks

The only category currently supported for this call is 8, which is for the logical disk. FSDs call *FSH_DEVIOCTL* to control device driver operation independently from I/O operations. This is typically in filtering *DosDevIOCtl* requests when passing the request on to the device driver.

An FSD needs to be careful of pointers to buffers that are passed to it from *FS_IOCTL*, and what it passes to *FSH_DEVIOCTL*. It is possible that such pointers may be real mode pointers if the call was made from the DOS mode. In any case, the FSD must indicate whether it initiated the *DevIOCtl* call, in which case the kernel can assume that the pointers are all protect mode pointers, or if it is passing user pointers on to the *FSH_DEVIOCTL* call, in which case the mode of the pointers will depend on whether this is the DOS mode or not. An important thing to note is that the FSD must not mix user pointers with its own pointers when using this helper.

FSH_DEVIOCTL may block.

Note: OS/2 does not validate input parameters. Therefore, an FSD should call *FSH_PROBEBUF* where appropriate.