

## FS\_OPENCREATE

### Purpose

Opens (or creates) the specified file.

### Calling Sequence

```
int far pascal FS_OPENCREATE(pcdfsi, pcdfsd, pName, iCurDirEnd, psffsi,
                             psffsd, ulOpenMode, usOpenFlag, pusAction,
                             usAttr, pcEABuf, pfggenflag)

struct cdfs_i far * pcdfsi;
struct cdfs_d far * pcdfsd;
char far * pName;
unsigned short iCurDirEnd;
struct sffsi far * psffsi;
struct sffsd far * psffsd;
unsigned long ulOpenMode;
unsigned short usOpenFlag;
unsigned short far * pusAction;
unsigned short usAttr;
char far * pcEABuf;
unsigned short far * pfggenflag;
```

### Where

*pcdfs\_i* is a pointer to the file-system-independent working directory structure.

The contents of this structure are invalid for direct access opens.

*pcdfs\_d* is a pointer to the file-system-dependent working directory structure. The contents of this structure are invalid for direct access opens. For remote character devices, this field contains a pointer to a DWORD that was obtained from the remote FSD when the remote device was attached to this FSD. The FSD can use this DWORD to identify the remote device.

*pName* is a pointer to the ASCII name of the file to be opened.

The FSD does not need to verify this pointer.

*iCurDirEnd* is the index of the end of the current directory in *pName*.

This is used to optimize FSD path processing. If *iCurDirEnd* == -1, there is no current directory relevant to the name text, that is a device. This value is invalid for direct access opens.

*psffsi* is a pointer to the file-system-independent portion of an open file instance.

*psffsd* is a pointer to the file-system-dependent portion of an open file instance.

*ulOpenMode* indicates the desired sharing mode and access mode for the file handle.

See OS/2 Version 2.0 Control Program Programming Reference for a description of the *OpenMode*

parameter for *DosOpen*.

An additional access mode 3 is defined when the file is being opened on behalf of OS/2, loaded for purposes of executing a file or loading a module. If the file system does not support an executable attribute, it should treat this access mode as open for reading. The value of *ulOpenMode* passed to the FSD will be valid.

*usOpenFlag* indicates the action taken when the file is present or absent.

See OS/2 Version 2.0 Control Program Programming Reference for a description of the *usOpenFlag* parameter for *DosOpen*.

The value of *openflag* passed to the FSD is valid. This value is invalid for direct access opens.

*pusAction* is the location where the FSD returns a description of the action taken as governed by *openflag*.

The FSD does not need to verify this pointer. The contents of *Action* are invalid on return for direct access opens.

*usAttr* are the OS/2 file attributes.

This value is invalid for direct access opens.

*pcEABuf* is a pointer to the extended attribute buffer.

This buffer contains attributes that will be set upon creation of a new file or upon replacement of an existing file. If NULL, no extended attributes are to be set. Addressing of this data area has not been validated by the OS/2 kernel (see *FSH\_PROBEBUF*). The contents of *EABuf* are invalid on return for direct access opens.

*pfgenflag* is a pointer to an unsigned short of flags returned by the FSD. The only flag currently defined is 0x0001 *fGenNeedEA*, which indicates that there are critical EAs associated with the file. The FSD does not need to verify this pointer.

## Remarks

For the file create operation, if successful, *ST\_CREAT* and *ST\_PCREAT* are set. This causes the file to have zero as last read and last write time. If the last read/write time stamps are to be the same as the create time, simply set *ST\_SWRITE*, *ST\_PWRITE*, *ST\_SREAD*, and *ST\_PREAD* as well.

For the file open operation, the FSD copies all supported time stamps from the directory entry into the SFT.

Note: ALSO NEW FOR 2.0, it is suggested that the FSD copy the DOS file attributes from the directory entry into the SFT. This allows the FSD and the OS2 kernel to handle FCB opens more efficiently.

The sharing mode may be zero if this is a request to open a file from the DOS mode or for an FCB request.

FCB requests for read-write access to a read-only file are mapped to read-only access and reflected in the *sfi\_mode* field by the FSD. An FCB request is indicated by the third bit set in the *sfi\_type* field.

The flags defined for the *sfi\_type* field are:

- type == 0x0000 indicates file.
- type == 0x0001 indicates device.
- type == 0x0002 indicates named pipe.
- type == 0x0004 indicates FCB open.
- All other values are reserved.

FSDs are required to initialize the *sfi\_type* field, preserving the FCB bit.

On entry, the *sfi\_hvpb* field is filled in. If the file's logical size (EOD) is specified, it is passed in the *sfi\_size* field. To the extent possible, the file system tries to allocate this much storage for efficient access.

Extended attributes are set for:

- the creation of a new file
- the truncation of an existing file
- the replacement of an existing file.

They are not set for a normal open.

If the standard OS/2 file creation attributes are specified, they are passed in the *attr* field. To the extent possible, the file system interprets the extended attributes and applies them to the newly-created or existing file. Extended attributes (EAs) that the file system does not itself use are retained with the file and not discarded or rejected.

When replacing an existing file, the FSD should not change the case of the existing file.

FSDs are required to support direct access opens. These are indicated by a bit set in the *sffsi.sfi\_mode* field. See OS/2 Version 2.0 Control Program Programming Reference for more information on *DosOpen*. Some of the parameters passed to the FSD for direct access opens are invalid, as described above.

On a successful return, the following fields in the *sffsi* structure must be filled in by the file system driver: *sfi\_size* and all the time and date fields.

The file-system-dependent portion of an open file instance passed to the FSD for *FS\_OPENCREATE* is never initialized.

Infinite FCB opens of the same file by the same DOS mode process is supported. The first open is passed through to the FSD. Subsequent opens are not seen by the FSD.

Any non-zero value returned by the FSD indicates that the open failed and the file is not open.

Note: This entry point's parameter list definition has changed from the 1.x IFS document. The *OpenMode* parameter has been widened from a unsigned short to a unsigned long. The upper word of the long is relevant only to a special SPOOLER FSD. For information about the upper word please contact the OS/2 Technical Interface group for the OEMI document for the 2.0 API

From:

<http://www.osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://www.osfree.org/doku/doku.php?id=en:ibm:ifs:routines:opncrea>

Last update: **2014/05/13 02:29**

