FS_NMPIPE

Purpose

Perform a special purpose named pipe operation remotely.

Calling Sequence

```
int far pascal FS_NMPIPE(psffsi, psffsd, OpType, pOpRec, pData, pName)

struct sffsi far * psffsi;
struct sffsd far * psffsd;
unsigned short OpType;
union npoper far * pOpRec;
char far * pData;
char far * pName;
```

Where

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.

OpType is the operation to be performed. This parameter has the following values:

NMP_GetPHandState	0x21
NMP_SetPHandState	0x01
NMP_PipeQInfo	0x22
NMP_PeekPipe	0x23
NMP_ConnectPipe	0x24
NMP_DisconnectPipe	0x25
NMP_TransactPipe	0x26
NMP_ReadRaw	0x11
NMP_WriteRaw	0x31
NMP_WaitPipe	0x53
NMP_CallPipe	0x54
NMP_QNmPipeSemState	0x58

pOpRec is the data record which varies depending on the value of *OpType*. The first parameter in each structure encodes the length of the parameter block. The second parameter, if non-zero, indicates that the pData parameter is supplied and gives its length. The following record formats are used:

```
union npoper {
    struct phs_param phs;
    struct npi_param npi;
    struct npr_param npr;
    struct npw_param npw;
    struct npq_param npq;
    struct npx_param npx;
    struct npp_param npp;
    struct npt_param npt;
    struct qnps_param qnps;
    struct npc_param npc;
    struct npd_param npd;
};

/* Get/SetPhandState parameter block */

struct phs_param {
    short phs_len;
    short phs_dlen;
    short phs_pmode; /* pipe mode set or returned */
};

/* DosQNmPipeInfo parameter block */
struct npi_param {
    short npi_len;
    short npi_dlen;
    short npi_level; /* information level desired */
};

/* DosRawReadNmPipe parameters */
/* data is buffer addr         */

struct npr_param {
    short npr_len;
    short npr_dlen;
    short npr_nbyt; /* number of bytes read */
};


/* DosRawWriteNmPipe parameters */
/* data is buffer addr          */

struct npw_param {
    short npw_len;
    short npw_dlen;
    short npw_nbyt; /* number of bytes written */
};

/* NPipeWait parameters */

struct npq_param {
    short npq_len;
    short npq_dlen;
    long npq_timeo;  /* time-out in milliseconds */
    short npq_prio;  /* priority of caller       */
};

/* DosCallNmPipe parameters */
/* data is in-buffer addr   */

struct npx_param {
    short npx_len;
    unsigned short npx_ilen;  /* length of in-buffer      */
    char far * npx_obuf;      /* pointer to out-buffer    */
    unsigned short npx_ilen;  /* length of out-buffer     */
    unsigned short npx_nbyt;  /* number of bytes read     */
    long npx_timeo;           /* time-out in milliseconds */
};

/* PeekPipe parameters, data is buffer addr */

struct npp_param {
    short npp_len;
    unsigned short npp_dlen;
    unsigned short npp_nbyt;   /* number of bytes read      */
    unsigned short npp_av10;   /* bytes left in pipe        */
    unsigned short npp_av11;   /* bytes left in current msg */
    unsigned short npp_state;  /* pipe state                */
};

/* DosTransactNmPipe parameters */
/* data is in-buffer addr       */

struct npt_param {
    short npt_len;
    unsigned short npt_ilen;  /* length of in-buffer   */
    char far * npt_obuf;      /* pointer to out-buffer */
    unsigned short npt_olen;  /* length of out-buffer  */
    unsigned short npt_nbyt;  /* number of bytes read  */
};

/* QNmPipeSemState parameter block */ /* data is user data buffer */

struct qnps_param {
    unsigned short qnps_len;  /* length of parameter block      */
    unsigned short qnps_dlen;  /* length of supplied data block */
    long qnps_semh;            /* system semaphore handle       */
    unsigned short qnps_nbyt;  /* number of bytes returned      */
};

/* ConnectPipe parameter block, no data block */

struct npc_param {
    unsigned short npc_len;   /* length of parameter block */
    unsigned short npc_dlen;  /* length of data block      */
};

/* DisconnectPipe parameter block, no data block */
```

```
struct npd_param {
    unsigned short npd_len;   /* length of parameter block */
    unsigned short npd_dlen;  /* length of data block      */
};
```

pData is a pointer to a user data for operations which require it. When the pointer is supplied, its length will be given by the second element of the *pOpRec* structure.

pName is a pointer to a remote pipe name. Supplied only for *NMP_WAITPIPE* and *NMP_CALLPIPE* operations. For these two operations only, the psffsi and psffsd parameters have no significance.

Remarks

This entry point is for support of special remote named pipe operations. Not all pointer parameters are used for all operations. In cases where a particular pointer has no significance, it will be NULL.

This entry point will be called only for the UNC FSD. Non-UNC FSDs are required to have this entry point, but should return NOT SUPPORTED if called.