Calling the .XPP Driver Sending commands to the server

The .XPP driver implements the workstation side of ASP and provides a mechanism for the workstation to send AppleTalk Filing Protocol (AFP) commands to the server.

Allocating Memory

Every call to the .XPP driver requires the caller to pass in whatever memory is needed by the driver for the call, generally at the end of the queue element. When a session is opened, the memory required for maintenance of that session (that is, the Session Control Block) is also passed in.

For standard **Device Manager** calls, a queue element of a specific size equal to IOQElSize is allocated. When issuing many calls to XPP, it is the caller's responsibility to allocate a queue element that is large enough to accommodate the .XPP driver's requirements for executing that call. Once allocated, that memory cannot be modified until the call complete.

Opening the .XPP Driver

To open the .XPP driver, issue a **Device Manager** **Open** call. The name of the .XPP driver is '.XPP'. The original Macintosh ROMs require that .XPP be opened only once. With new ROMs, the .XPP unit number can also be obtained through an **Open** call. With old ROMs only, the .XPP unit number must be hard coded to XPPUnitNum (40) since only one **Open** call can be issued to the driver.

The .XPP driver cannot be opened unless AppleTalk is open. The application must ensure that the .MPP and .ATP drivers are opened.

The xppLoaded bit (bit 5) in the PortBUse byte in low memory indicates whether or not the .XPP driver is open.

```
// Procedure to open the .XPP driver
// Routine: OpenXPP
// Assuming inclusion of <MacHeaders>

// Open the .XPP driver and return the driver refNum for it.

// Exit: d0 = error code (ccr's set)
//   d1 = XPP driver refNum (if no errors)

// All other registers preserved

#include <AppleTalk.h>
#include <SysEqu.h>
#include <Traps.h>

#define xppUnitNum  40 // default XPP driver number
#define xppTfRNum   -(xppUnitNum+1)    // default XPP driver refNum

#define ioQElSize 50

#define ioFileName    0x12        // file name pointer [pointer]
```

```
#define ioPermssn     0x1B        // permissions [byte]
#define ioRefNum      0x18        // file reference number [word]

void openXPP (void);

void openXPP ()
{
    char XPPName[6] = "\p.XPP";    // driver name

    asm {
        movem.l    a0-a1/d2, -(a6) ;save registers
        move    ROM85,d0      ;check ROM type byte
        bpl.s    @10            ;branch if >=128K ROMS
        btst     #xppLoadedBit,PortBUse ;is the XPP driver open already
        beq.s    @10    ;if not open, then branch to Open code
        move    #xppTfRNum,d1   ;else use this as driver refnum
        moveq  #0,d0 ;set noErr
        bra.s    @90          ;and exit

        ; XPP driver not open.  Make an _Open call to it. If using a 128K
        ; ROM machine and the driver is already open, we will make another
        ; Open call to it just so we get the correct driver refNum.

@10:    sub #ioQElSize,a6 ;allocate temporary param block
        move.l a6,a0              ;a0 -> param block
        lea  XPPName, a1  ;a1 -> XPP (ASP/AFP) driver name
        move.l a1, ioFileName(a0)    ;drivername into param block
        clr.b    ioPermssn(a0)    ;clear permissions byte
        dc.w    _Open
        move    ioRefNum(a0),d1 ;d1 = driver refnum (invalid if error)
        add #ioQElSize, a6          ;deallocate temp param block
@90:    movem.l (a6)+,a0-a1/d2     ;restore registers
        tst  d0                   ;error? (set ccr's)

    }
}
```

From a high-level language, XPP can be opened through the **OpenXPP** call, which returns the driver's reference number.

Open Errors

Errors returned when calling the **PBOpen** routine in the **Device Manager** if the function does not execute properly include the following:

- errors returned by System
- portInUse (-97) is returned if the AppleTalk port is in use by a driver other than AppleTalk or if AppleTalk is not open

Closing the .XPP Driver

To close the .XPP driver, call the **PBClose** routine in the **Device Manager**.

Warning: There is generally no reason to close the driver. Use this call

sparingly, if at all. This call should generally be used only by system-level applications.

Close Errors

Errors returned when calling the **PBClose** routine in the **Device Manager** if the function does not execute properly include the following:

- errors returned by the System
- closeErr (-24) (new ROMs only) is returned if you try to close the driver and there are sessions active through that driver. When sessions are active, closeErr is returned and the driver remains open.
- on old ROMs the driver is closed whether or not sessions are active and no error is returned. Results are unpredictable if sessions are still active.

Session Control Block

The Session Control Block (SCB) is a nonrelocatable block of data passed by the caller to XPP upon session opening. XPP reserves this block for use in maintaining an open session. the SCB size is defined by the constant scbMemSize. The SCB is a locked block, and as long as the session is open, the SCB cannot be modified in any way by the application. There is one SCB for each open session. This block can be reused once a **CloseSession** call is issued and completed for that session or when the session is indicated as closed.