New Technical Notes

Macintosh

Developer Support

Printer Access Protocol Q&As
Networking

Revised by: Developer Support Center   October 1992
Written by: Developer Support Center   October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

Monitoring PAP packets with AppleTalk Peek
Date Written: 8/7/90
Last reviewed: 8/30/90

Are there hardware or software tools for monitoring AppleTalk Printer Access Protocol (PAP) packets in some direct way?

———

A tool called "AppleTalk Peek" allows you to monitor packets that are being sent over AppleTalk. It is available on the AppleLink and on the latest *Developer CD Series* disc. You can find the tool via the following path: Storage and Communication Folder : Network Folder : AppleTalk Tools Folder. This tool might enable you to determine experimentally, PostScript handle sizes that work best for your application.

Using PAP & code for finding printer driver under System 7
Date Written: 6/6/91
Last reviewed: 8/1/91

Is there any documentation or sample code on the Printer Access Protocol (PAP), besides

Inside AppleTalk and MacTutor's sample? Or is there a better way to shove 1 MB and larger files down the pipe to a PostScript device?

____

The question that you ask has become a popular one these days; it seems that a lot of people are writing PAP programs. Unfortunately, there isn't really a clean-cut answer. There are several options:

1) The code that appeared in MacTutor seems to work just fine. Of course, it is unsupported by Apple and it may very well break under a future system software release. It is definitely the quickest way to get a PAP-based program coded.

2) You could write your own PAP interfaces. This is a very time consuming operation, but it also most likely not to break under future systems. Other companies have done it, and it is fairly straightforward.

3) Through software licensing, you can license a library named papworkstation.o. It is a library only, with little or no documentation, and it is unsupported. It is, however, used by several third parties, and it works OK.

In case you do use the MacTutor code, the following example shows a method for finding the printer driver under System 7:

```
FUNCTION GetPAPDriver: BOOLEAN;
VAR
 theResFile:         INTEGER;
 theWorld:         SysEnvRec;
 theError:         OSErr;
 DoinFolders:     BOOLEAN;
 myFeature:         LONGINT;
 sysVRefNum:         INTEGER;
 sysDirID:         LONGINT;
 gotDriver:         BOOLEAN;

 numStr:           Str255;
BEGIN
 gotDriver := TRUE;
 DoinFolders := FALSE;

(*-------------------- The 7.0 Way!  ------------------------*)
 IF GestaltAvailable THEN BEGIN
   theError := Gestalt(gestaltFindFolderAttr, myFeature);
   IF theError = noErr THEN BEGIN
     DoinFolders := BitTst(@myFeature, 31-gestaltFindFolderPresent);
     IF DoinFolders THEN BEGIN
        theError := FindFolder(kOnSystemDisk, kExtensionFolderType,
                               kDontCreateFolder, sysVRefNum, sysDirID);
        IF theError <> noErr THEN BEGIN
           DebugStr('FindFolder err');
           gotDriver := FALSE;
        END;
     END;
   END;
 END;

 (* If FindFolder was not available, use good ol' SysEnvirons (thanks Jim)*)
 IF NOT DoinFolders THEN BEGIN
   (* No FindFolder!  Gotta do it the old way... *)
   theError := SysEnvirons(1, theWorld);
   IF theError = noErr THEN
     sysVRefNum := theWorld.sysVRefNum
```

```
    ELSE
      gotDriver := FALSE;
 END;
(*----------------------------------------------------------------*)

 (* Okay, at this point we should have found the folder where the drivers *)
 (* are. If we  are running under System 7.0, this folder will be the     *)
 (* extensions folder in the System folder. If we are pre-7.0, this will  *)
 (* be the System folder. If FindFolder was available, then we have to    *)
 (* use HOpenResFile because FindFolder uses dirIDs. If we just used       *)
 (* SysEnvirons (ie. pre 7.0), then we can use OpenRFPerm.                 *)
 IF gotDriver THEN BEGIN
   IF DoinFolders THEN
     theResFile := HOpenResFile(sysVRefNum, sysDirID, driverName, fsRdPerm)
   ELSE
     theResFile := OpenRFPerm(driverName, sysVRefNum, fsRdPerm);

   theError := ResError;
   IF theError = noErr THEN BEGIN
     driverCode := GetResource('PDEF', 10);
     HNoPurge(driverCode);
     DetachResource(driverCode);
     IF ResError = noErr THEN BEGIN
       LWName := StringHdl(GetResource('PAPA', -8192));
       DetachResource(Handle(LWName));
       CloseResFile(theResFile)
     END ELSE
         gotDriver := FALSE;
   END ELSE BEGIN
     NumToString(theError, numStr);
     DebugStr(numStr);
     gotDriver := FALSE;
   END;
 END ELSE
   gotDriver := FALSE;
 GetPAPDriver := gotDriver;
END; (* GetPAPDriver *)
```