

New Technical Notes

Macintosh



®

Developer Support

Printer Driver Q&As

Imaging M.IM.PrtDvr.Q&As

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.

|New Q&As and Q&As revised this month are marked with a bar in the side margin.

Where to get documentation on writing a Macintosh printer driver

Written: 4/9/91

Last reviewed: 8/1/92

Where can I find documentation on how to write a Macintosh printer driver equivalent to the ImageWriter or LaserWriter driver? In particular, how are Printing Manager and QuickDraw commands translated into calls to the printer driver?

—

DTS's "Learning to Drive" document and "StdFileSaver" source code, available in AppleLink's Developer Support folder and on the latest Developer CD Series disc, are helpful references.

Writing a Macintosh printer driver

Written: 5/3/89

Last reviewed: 8/1/92

I have a printer I would like to connect to the Macintosh. Where can I find information on writing a printer driver?

—

A Macintosh printer driver is more than a standard device driver. A printer driver contains code to implement all of the standard Macintosh Printing Manager routines. The code includes

routines like PrOpen/PrClose, and PrOpenDoc/PrCloseDoc. A Printing Resource File contains all of the resources (including code) to implement the Macintosh Printing Manager for a particular device. A driver works best with all Macintosh applications if it supports both the high and low-level Printing Manager interfaces, as well as the PrGeneral (IM V:410) routine and its associated opcodes.

Apple is not currently supporting the development of Chooser-selectable device drivers, at least not those that support printers. There are many reasons for this, and here are at least a few of them:

First, there are no documentation or examples available. Each of the Apple printer drivers is unique. They are all written almost entirely in 68000 assembler, and consequently, are not very easy to read. Since each driver is different, the only real documentation for how the drivers work is the source code to the driver. There is some general interface information available, but important information, like how the driver manages its print record, is described only in the source code. This information could be extracted and distilled into some kind of document, but this brings up the next problem:

The Macintosh Printing Manager is currently being revised and enhanced. These revisions will require major changes to the architecture of the Printing Manager. It's not clear what effect these changes will have on existing printer drivers, but it is very possible that the drivers that exist today will have to be revised significantly to run under the new Printing Manager.

Apple is designing and implementing the next generation Macintosh Printing Manager. Developers' suggestions for features are being taken into account. One of the goals of the new architecture is to make writing drivers much simpler, and to allow the sharing of code between drivers. When this architecture becomes available, Apple will reconsider its position of driver development, and will probably end up with some kind of licensing agreement for developers that want to write drivers. Until then, the preceding reasons should be adequate justification for NOT attempting a Macintosh printer driver at this time.

If you still aren't convinced, and want to write a driver despite the challenge, see the article in the December '88 issue of MacTutor magazine. There is an example printer driver written in C. It is just a skeleton driver, and does not handle any of the more difficult problems such as line layout, font substitution, graphics, or banding. But it is a start, and gives you a general idea of the structure of a driver. Since some developers began writing their drivers before Apple discontinued support, DTS is still answering questions concerning driver development. However, these questions are limited to those that can be answered by the DTS engineers. You should also be aware that some of the techniques used by Apple printer drivers are considered proprietary. Information on methods used for line layout as well as certain aspects of font handling will not be disclosed.

X-Ref:
Printing Manager

Changing printer driver settings without using the dialog

Written: 11/6/91

Last reviewed: 8/1/92

To change default printer driver settings without using the dialog, keep a copy of the print record with the various settings you want, and when you want to choose your settings, just use your special print record. The current Macintosh print architecture doesn't allow procedural access to the print record, so you must call the dialogs once for yourself, choose your settings,

and then save a copy of the print record that is produced. Some settings are job-dependent, but values that are retrieved from the print record can be changed using this technique. You can only save the settings in the Style dialog. Call PrValidate when retrieving a print record from the resource fork to make sure it is “good.”

See *develop*, issue #1, pages 58-65, for a complete discussion of changing default printer driver settings and sample code.

Macintosh print record wDev field

Written: 5/3/89

Last reviewed: 8/1/92

What is the Macintosh print record wDev value for the ImageWriter LQ, LaserWriter IISC, and AppleFax Modem?

Apple strongly discourages the use of the wDev field in the print record for several reasons. First, this field contains a unique ID for each printer driver on the Macintosh. Coding your application to use this ID makes it device dependent, and device dependence will cause many problems in the future as the Macintosh Printing Manager continues to evolve. Many applications currently check for wDev = 3 to determine whether or not to send PostScript. Although the Apple LaserWriter driver has a wDev of 3, third party printer drivers for PostScript devices do not, so if you're application determines whether or not to send PostScript based on wDev alone, your application may incorrectly print with QuickDraw on third party PostScript devices. A second problem concerns spoolers and spool files. If a spooler is installed between an application and the printer driver that is going to do the printing, the application receives the wDev ID of the spooler instead of the target driver. If the application makes assumptions based on this ID, it probably gets unexpected results. In the future, it may also be possible to redirect spool files. This means that a file that was originally spooled for a PostScript printer may be redirected to a QuickDraw device. If the spool file contains only the PostScript representation of the document, it will be useless to the QuickDraw device. Despite these strong warnings, some developers are convinced they need the wDev values, and there may be some vertical applications where they're needed. For those cases, here is the current list of wDev IDs for Apple printer drivers:

Device	wDev (Hi Byte)
ImageWriter I/II:	1
LaserWriter, LaserWriter Plus, LaserWriter IINT,	
LaserWriter IINTX:	3
LaserWriter IISC:	4
ImageWriter LQ:	5
AppleFax Modem:	10

DTS does not support the use of these constants. They are definitely subject to change.

Determining if a printer driver accepts Color QuickDraw calls

Written: 11/21/90

Last reviewed: 8/1/92

How do you find out if a printer driver accepts Macintosh Color QuickDraw calls?

—

Check to see if the printer driver has returned a color GrafPort to your application after the call to PrOpenDoc (that is, the port that PrOpenDoc returns).

To determine if the GrafPort is color, you need to check to see if rowBytes from the GrafPort are less than 0. The following code fragment demonstrates this idea:

```
(* This function determines if the port passed to it is a color port. If *)
(* so, it returns TRUE. *)

FUNCTION ColorPort(portInQuestion: GrafPtr): BOOLEAN;
BEGIN
  IF portInQuestion^.portBits.rowBytes < 0 THEN
    ColorPort := TRUE
  ELSE
    ColorPort := FALSE;
END;
```

Printing in mixed Macintosh System 6/7 network

Written: 12/11/90

Last reviewed: 8/1/92

What is the recommended printer driver for a network environment with mixed operating systems (such as System 6.0.5 on a Macintosh Plus and 7.0 on two Macintosh IIfx systems)?

In a mixed 6.0.x/7.0 network we recommend upgrading all systems to the 7.0 print drivers—even systems running 6.0.x. The 7.0 LaserWriter drivers work fine under system 6.0.x and will avoid any printer re-initialization problems. Use the 7.0 printing disk (or the printing install folder), and launch the Installer from that disk. It has scripts needed to install 7.0 print drivers on a 6.0.x system. Do *not* select the Installer options for printers from the main 7.0 Installer, because these scripts currently are only for systems running 7.0.

If a Personal LaserWriter NTR is used on the network, you must upgrade all workstations to the 7.1.1 LaserWriter driver or later. (The NTR requires at least LaserWriter 7.1.1.)

LaserWriter 6.x driver pixel map bug fixed

Written: 11/1/90

Last reviewed: 8/1/92

When printing PixMaps in color in 16- or 32-bit modes, the Macintosh LaserWriter driver 6.0 or 6.0.1 generates a PostScript error. Why?

The color versions of the LaserWriter driver (6.0 and 6.0.1) have a bug that prevents them from printing any images deeper than 8 bits. The bug has been fixed in the 6.1 version of the driver, but unless the newer driver is available, you must depth convert your images to 8 bits

before passing them to the Printing Manager. Although there is a lot of overhead involved here, you don't actually lose any information. The PostScript “image” operator can handle depths only up to 8 bits anyway. Letting QuickDraw convert the image from 16/32 to 8 bits will cause dithering, which will make the image look nicer than if it was just an 8-bit image all along.

System 7.0 LaserWriter Driver & choosing nonstandard page sizes

Written: 4/8/91

Last reviewed: 8/1/92

With the System 7.0 version of the LaserWriter Driver, when the user selects Envelope from the Page Setup dialog, the page size returned by the driver is still a standard page: 8.5 x 11. How do you recommend that applications display the page size when the user has chosen a nonstandard page size?

—

We recommend that you have the page preview show a full page instead of an envelope-sized page. The LaserWriter Driver supports a large number of PostScript devices, and it can't be sure whether the envelope will be fed on the right, left, or center of the paper tray. If you show the full page, a user can print on any device by putting the text in the correct location for that device.

Manufacturers of PostScript® printers can add custom page sizes to the LaserWriter Driver. If they do, the representation on the screen will be whatever they decide to define. Applications should not try to interpret custom page sizes. If your application ignores the results returned by the driver, you risk incompatibility down the road.

Asynchronous LaserWriter driver no longer supported

Written: 9/24/91

Last reviewed: 8/1/92

What is causing incorrect characters to be printed on the Asynchronous LaserWriter driver we licensed from Apple for use with a non-Apple PostScript-equipped printer?

—

The printing of incorrect characters is probably due to an incompatibility between the async driver and the version of PostScript being used in the printer.

The reason for the incompatibility is that the driver is no longer supported and hasn't been revised for quite some time. The driver was originally developed outside of Apple, and Apple licensed the driver for its use. The company subsequently went out of business and the driver hasn't been revved since.

PostScript and Apple's AppleTalk LaserWriter drivers have been updated and changed quite a bit since then, which is why the characters print correctly when using AppleTalk, but not when printing asynchronously. The old Async driver just doesn't know how to handle the new PostScript versions and therefore prints "garbage" characters.

The SL Laser II driver is no longer supported by Apple, and at this time Apple has no plans to update this driver or write a new one. Since things work correctly with AppleTalk, you

should confine usage to AppleTalk when printing PostScript to the LaserWriter. If this is unacceptable, you might check around with different clearinghouses to see if a third-party developer has a compatible, up-to-date Asynchronous LaserWriter driver that you can use.

Use LaserWriter Driver srcCopy instead of srcOr transfer mode

Written: 5/1/91

Last reviewed: 8/1/92

I'm creating PICTs that are comprised of many lines drawn in srcOr mode. When using the LaserWriter 6.x or 7.x driver with the Color/Grayscale radio button selected, some lines fail to print. Why is this happening?

—

The problem is a bug in the LaserWriter driver. Sometimes, when using a CGrafPort, the driver doesn't reproduce lines drawn in srcOr mode. (A CGrafPort is used when the Color/Grayscale print option is selected; in Black & White print mode, a regular grafPort is used.) A workaround is to use srcCopy instead of srcOr when drawing QuickDraw objects within your PICTs.

Save and restore long word if using \$948 under System 7

Written: 6/11/91

Last reviewed: 8/1/92

Unless my Macintosh application restores the contents of the long word at \$948 after using that space for globals, the Finder draws icons incorrectly and my third-party LaserWriter driver crashes. Is somebody now using \$948 for other purposes?

—

This is a known System 7 icon drawing bug, and also a bug with the printer driver that you are using.

The icon drawing utilities are trying to determine if a print page is currently open, by looking at that variable (\$948 is part of printvars). This turns out not to be such a good idea, and it will be fixed in the next release of the system software.

The printer driver bug that you are experiencing is that every printer driver must return the variable at \$948 to -1 when they are done printing. Also, while printing, the driver should set it to <> -1.

LaserWriter 6.1 versus 7.0 (use new installer for both)

Written: 8/15/91

Last reviewed: 8/1/92

What is the difference between the LaserWriter driver version 6.1 and 7.0?

—

The 6.1 driver is basically an early release of the 7.0 driver. A number of bugs were corrected in the final 7.0 release. In terms of functional differences, there aren't any. Some features just function a little bit more like they're supposed to in the 7.0 version.

Since the 7.0 driver can be used with 6.0.x systems, there's really no reason to use the 6.1 driver anymore. If you decide to use the 7.0 driver with a 6.0.x system, be sure to install it with the 7.0 installer. Several system files must be removed or updated to work with the 7.0 driver and the installer correctly handles this.

Update Backgrounder if using LaserWriter 7.0 under System 6

Written: 7/10/91

Last reviewed: 8/1/92

When we use the LaserWriter driver 7.0 with System 6.0, the Chooser's Background Printing "On" button is always dimmed. Is there any way to enable background printing?

—

To get the LaserWriter 7.0 driver to work with System 6.0.x, update your Backgrounder file with the version on the Printing Tools Disk that's used to install System 7.0.

In short, to use the 7.0 LaserWriter Driver, 7.0 PrintMonitor, and System 6.0.x, place these files from the System 7.0 Printing Tools installation disks into your System Folder. All three must be used together for printing to work correctly. After you've placed these files in the System Folder, you should find the Background Printing option enabled for your use.

There are no known incompatibilities with the 7.0 drivers and System 6.0.x, so everyone should use the latest drivers, even with a mixed environment. This will also get rid of "printer wars" that occur when users use more than one version of the LaserWriter driver to print to the same printer. (The printer must be reinitialized if the current user uses a different version of the driver than what was used by the previous user.)

LaserPrep 7.0 file & AppleShare Print Server

Written: 7/15/91

Last reviewed: 8/1/92

The LaserPrep 7.0 file is included on the System 7 "Printing" disk and the System 7 Group Update CD for upgrading the AppleShare Print Server to support the LaserWriter 7.0 driver. The AppleShare Print Server has its own LaserWriter driver built in and all it needs to print is a LaserPrep file. In fact, the AppleShare Print Server completely ignores any LaserWriter driver installed in the System Folder of the server.

Page 49 of the "System 7 Group Upgrade Guide" states the following procedure for upgrading an AppleShare Print Server to support the System 7 LaserWriter drivers:

1. Shut down the print server software.
2. Install the printer driver update...
3. Drag the LaserPrep icon from the Printer Update folder to the Server Folder of the print server Macintosh.
4. Restart the print server Macintosh and the AppleShare Print Server software.

Actually, step #2 in the "System 7 Group Upgrade Guide" is unnecessary. Installing the printer driver update has no effect on the AppleShare Print Server software.

LaserWriter 7.0 driver and LaserPrep dictionary

Written: 8/29/91

Last reviewed: 8/1/92

In the old LaserWriter drivers it was possible to create a PostScript file with or without the LaserPrep dictionary (“f” or “k” key). Is it possible to generate a file without the dictionary with the LaserWriter 7.0 driver?

—

In 7.0 printing, a LaserPrep dictionary is always sent with a print job. This is done to prevent constant reinitialization of the LaserWriter by conflicting printer drivers. You cannot prevent it from being sent. Fortunately the 7.0 version of the LaserPrep dictionary is much smaller ($\approx 40\text{K}$ total) than its 6.x predecessors.

LaserWriter ignores ForeColor while filling smoothed polygon

Written: 10/8/91

Last reviewed: 8/1/92

When doing the FillRgn for drawing the fill of a smoothed polygon (as described in Macintosh Tech Note #91) the foreColor isn't used on the LaserWriter. Any way to make it work?

—

You've discovered a design limitation of the LaserWriter driver. Things that have patterns associated with them are rendered by using the LaserWriter screen operators. This results in an assumption of the foreground color being black and the background color being white, which is what's causing the problem you noticed. In short, it makes the driver ignore your foreground and background colors.

You can work around the problem by working in an offscreen GWorld first and then CopyBitsing everything to the printer port using srcCopy. There are a couple of problems with this approach: First, don't do it with text or your text will be turned into a bitmap and you'll get the "jaggies." Second, you'll probably want to increase the printer port's resolution from its default of 72 dpi for better results. A value of 288 dpi works nicely since it's an even multiple of QuickDraw's native 72-dpi resolution. Also, make sure that the GWorld you create has the same bounds as the printer port's rPage rectangle to avoid unnecessary scaling and clipping. After you've done all that, draw into it and CopyBits the result to the printer port. The nice thing about doing everything offscreen first is that then you can use some of the non-printer-friendly transfer modes like blend or dithered. Also, you can use ForeColor/BackColor and get the right thing this way.

If you don't want to use this method for all printers, (since it can be quite a memory hog), you can check for the LaserWriter driver and use this method in just that case. For other drivers, you should just print as normal.

So, what do you need to do all this? Well, to set the resolution of the printer port, use PrGeneral as described in Inside Macintosh Volume V and develop #3.

To determine whether you have the LaserWriter driver or not, check the high byte of the wDev field in your print handle. This is described in the Tech Note "Optimizing For The LaserWriter—Techniques." While this method might break some day, it's currently the best way to determine which driver you're using, and Apple will have to let developers know before we break it.

If the high byte of the wDev is 3, then you either have the LaserWriter driver or a third-party driver impersonating the LaserWriter. Some PostScript drivers do that because many apps assume a wDev of 3 means a PostScript printer, and anything else doesn't. By acting like Apple's LaserWriter driver, they get preferential treatment from apps that "special case" for PostScript. That's not really a problem in this situation.

LaserWriter driver PostScript error strings & document names

Written: 12/11/91

Last reviewed: 8/1/92

I discovered an interesting bug in the Macintosh's LaserWriter driver. If the word "timeout" is in the name of a document, the LaserWriter driver will give a timeout error -8132. Are there similar magic words?

—

PostScript error messages are sent from the LaserWriter to the driver as text streams. The driver must check these strings to see if they contain an error message. If a document is named something that contains the same string as a PostScript error message, the driver thinks there is an error when the printer sends the "status: printing document XXXXX" message.

Other strings cause similar problems; one of them is "printer out of paper." If you want to see the rest of the strings, take a look at the LaserWriter printer driver resource type 'PREC' ID = 109. That is where the rest of the error strings are.

Until this is fixed, be careful about putting PostScript error messages into the names of your documents, because the driver will choke every time.

Personal LaserWriter NTR has longer product string

Written: 3/17/92

Last reviewed: 8/1/92

The following line of my PostScript code causes the Personal LaserWriter NTR to gag, but it has worked with all other Apple LaserWriter printers up to now:

```
statusdict/product get str cvs show % Gets & prints the name of the printer
```

where str is defined as:

```
/str 20 string def
```

—

The product name stored for this printer is (LaserWriter Personal NTR) or (Personal LaserWriter NTR), depending on whether you're using statusdict or systemdict, respectively. In either case, the product's length is 24 characters. Since you're only allocating 20 characters for the 2nd string you use with cvs, you're getting a rangecheck error. Changing str to:

```
/str 24 string def
```

fixes the problem. You may want to make the string even larger than 24 characters to accommodate longer product names in the future.

Assumptions about the length of PostScript product strings are a common problem with new printers (having more verbose names). In fact, you should have the same trouble if you run your original PostScript on the Personal LaserWriter NT, which has a 23 character name.

LaserWriter 5.x and low-level Printing Manager interface

Written: 5/3/89

Last reviewed: 8/1/92

When I try to cancel a job using LaserWriter driver 5.0, 5.1, or 5.2 my Macintosh crashes. Why?

—

The 5.x versions of the LaserWriter driver have a bug that affects applications using the low-level Printing Manager interface. Attempting to cancel a job will cause the Printing Manager to generate an “Address Error”. This problem is fixed in the 6.0 and later releases of the LaserWriter driver. There is no workaround for the 5.x versions.

LaserWriter 5.2 driver and DrawString problem

Written: 10/19/90

Last reviewed: 8/1/92

With Macintosh LaserWriter Driver 5.2, I can't use DrawString to print a string of length one, enclosed in quotes drawn by DrawChar. Even a character of value \$0A crashes the system.

—

This is a bug which has been fixed in LaserWriter Driver version 7.0. It appears that you need to watch for 0x0A unless you are printing to LaserWriter driver version 7.0. The clean way to check for LaserWriter driver version 7.0 is to check the version number of the driver via a call to PrDrvVrs (see IM II: 163) and make sure you have the right driver via the wDev. This approach is clean because you are looking for a particular driver version and not for a PostScript printer. This approach works for both System 6 and 7.

LaserWriter II SC fonts and 4x bitmap size

Written: 11/17/89

Last reviewed: 8/1/92

Why do I need to have four times the bitmap size of the font I want to print in on the LaserWriter II SC?

—

LaserWriter II SC fonts are four times the point size of the associated Macintosh screen font. The pixels on the screen are four times as far apart (center to center) as the dots printed by the LaserWriter SC. When the bitmaps for one of the printer fonts is printed, the result is a font with a resolution four times greater than that of the font displayed, but at a size identical to the font displayed on the screen.

Font families shipped with LaserWriters

Written: 11/17/89

Last reviewed: 8/1/92

What fonts and sizes are shipped with the LaserWriter Plus, LaserWriter II NT, LaserWriter II NTX, and LaserWriter SC?

—

The LaserWriter Plus, LaserWriter II NT, and LaserWriter II NTX shipped with a total of 11 font families with the sizes indicated below:

Times, Helvetica, Courier, Symbol: 9, 10, 12, 14, 18, 24

Palatino, Helvetica Narrow, ITC Bookman,
ITC Avant Garde Gothic, ITC Zapf Chancery,
ITC Zapf Dingbats, New Century Schoolbook 10, 12, 14, 18, 24

A total of four font families are shipped with the SC: Courier, Symbol, Times, and Helvetica in the following sizes: 9, 10, 12, 14, 18, 24, 36, 48, 56, 72, 96.

Disable “Graphics Smoothing” for printing large bitmap images

Written: 5/3/89

Last reviewed: 8/1/92

When printing large (possibly scanned) Macintosh bitmap images, the page is printed with small lines running horizontally through the image. Why?

—

In System 6.0 the QuickDraw DrawPicture call was revised to fix some problems. One of these problems concerned large bitmaps. To help solve the problem of bitmaps that were too large to be printed, DrawPicture was modified to “band” the CopyBits request if it was too large. Banding is the process of converting a large bitmap into several smaller bitmaps. This banding usually occurs vertically down the page. When DrawPicture bands a large bitmap into pieces, the smoothing algorithm of the LaserWriter is applied to each piece separately, instead of being applied to the entire bitmap at once. Since the process of smoothing involves removing some pixels, hairlines will be created between the bitmap bands. There is no way to tell the LaserWriter driver that the smaller bitmaps are all part of one large bitmap, so the only solution to this problem is to disable Graphics Smoothing when printing large bitmap images.

Why Macintosh System 6.0.4 has LaserWriter driver 5.2, not 6.0

Written: 11/17/89

Last reviewed: 8/1/92

Why are the LaserWriter 5.2 drivers on Macintosh System 6.0.4 and later instead of LaserWriter 6.0?

—

System 6.0.4 was designed only to provide support for the Macintosh IIfx and Macintosh Portable and not to provide any increased functionality to the Macintosh product line that earlier System 6 releases did not provide. LaserWriter 6.0 is still available on the AppleColor Disk which also includes version 1.0 of 32-Bit QuickDraw.

Difference between LaserWriter 7.0 and 7.1 Namer

Written: 3/3/92

Last reviewed: 8/1/92

Apple Software Licensing's "Exhibit C" lists localized Namer 7.0 versions in several languages, and mentions that these localized versions of the Namer are available in version 7.1. What is the difference in the Namer between 7.0 and 7.1?

—

The main difference between the LaserWriter utility 7.0 and 7.1 is that part of the Namer has been integrated into the 7.1 version of the software. Until the LaserWriter utility 7.1 was made available, the Namer was a standalone application. However, the LaserWriter utility does not do everything the Namer does. For example, the Namer renames AppleTalked ImageWriters, but the LaserWriter utility doesn't. For all your printer configuration needs, you should use the 7.1 version of the software.

ImageWriter and printing multiple copies in draft mode

Written: 1/6/92

Last reviewed: 8/1/92

Our application can't print multiple copies on the ImageWriter in Draft mode. If we type in "3" copies in the ImageWriter driver dialog in Draft mode, we only get one copy. Everything prints fine in Best or Faster modes. Is this a bug in the ImageWriter driver?

—

Although it seems like this must be a bug in the ImageWriter driver, it's not. In Draft mode, the application that's printing must make sure the required number of pages are printed. This involves cycling through the app's print loop for each copy to be printed.

In Draft mode on an ImageWriter, nothing is spooled to disk; the data is immediately sent to the printer. Therefore, the data is no longer available once it's sent to the printer (and there's no way for the print driver to resend it for multiple copies). When the ImageWriter spool-prints, the file that's created is printed the required number of times for you. Therefore, your app only needs to handle multiple copies when printing in Draft mode.

Here's what your app should do:

1. Validate the print record and present the job dialog. This allows the user to choose how many copies to print and allows the printer driver to adjust the print record to reflect how many copies your program has to print.
2. Get the number of copies that you're expected to handle from the print record. In the case of spool printing, this number will be set to 1 (since the multiple copies are handled for you by the printer driver). In any case, the following Pascal code will give you the correct info.

```
numCopies:= printHdl^.prJob.iCopies; (* From IM 2, pg. 151. *)
```

3. For each copy in numCopies, loop through the PrOpenDoc/PrCloseDoc section of your code.

This method is demonstrated in the code for the Macintosh Tech Note "A Printing Loop that Cares..." which describes a model print loop.

StyleWriter printer driver SetOrigin bug and workaround

Written: 12/12/91

Last reviewed: 5/21/92

SetOrigin works nicely for shifting an image slightly on the LaserWriter and ImageWriter, but does nothing with the 7.0 StyleWriter driver. If I put the SetOrigin call **after** PrOpenPage (which Inside Macintosh Volume II, page 156, tells me specifically **not** to do), it appears to work OK. What's the deal?

You're correct, there is a known bug with SetOrigin in StyleWriter driver versions 7.0 and 7.1. The bug has been fixed as of version 7.2.2 of the StyleWriter driver. Your program can work around the bug by checking if it is using an old driver version and then calling SetOrigin after PrOpenPage. Below is the code necessary to do this:

```
{ IsBuggyStyleWriter returns true if the printer is a StyleWriter with the
wrong version of the driver. It returns false if the printer is not a
StyleWriter, or the version number is OK }
```

```
FUNCTION IsBuggyStyleWriterDriver (tpr: THPrint): Boolean;
BEGIN
  IsBuggyStyleWriterDriver := false;
  { Ensure the prStl.wDev fld is initialized }
  { may not be necessary, if you are sure the THPrint passed is VALID }
  PrValidate(tpr);
  { StyleWriter device number of 101 in high byte of wDev}
  IF BSR(tpr^.prStl.wDev, 8) = 101 THEN
    { StyleWriter driver version 7.0 and 7.1 have bug }
    IF PrDrvrVers <= 71 THEN
      IsBuggyStyleWriterDriver := true;
END;
```

```
.....printing loop
{ This is a chunk of the printing loop where you would normally call
SetOrigin followed by PrOpenPage }
```

```
IF IsBuggyStyleWriterDriver(thePrRecHdl) THEN
BEGIN
  { This driver is buggy }
  PrOpenPage(thePrPort, nil);
  SetOrigin(-75, 0);
END
ELSE
  { This driver is not }
BEGIN
  SetOrigin(-75, 0);
  PrOpenPage(thePrPort, nil);
END;
```

```
.....Printing loop continues
```

The Printing Manager is supposed to be printer independent. But sometimes it's necessary to do some printer-specific things. Be cautioned that checking the high byte of the prStl.wDev field for a printer type is a little bit risky, because Apple has always maintained that it may change someday. In fairness, this technique is widely used by developers, so Apple will give plenty of warning before breaking it. Basically, the message is: "It works now, but be prepared to update your code someday."

Please see the Macintosh Technical Note "Optimizing For The LaserWriter—Techniques" for information on checking the prStl.wDev field to determine the printer type.

Dogcow logo is trademarked

Written: 9/10/92

Last reviewed: 9/15/92

We would like to use "dogcow" icon in our Page Setup dialog. Is the dogcow trademarked, and are there any restrictions on using this icon in our software?

The official word from Apple's Legal Group is yes, the dogcow logo is a trademark of Apple and is proprietary. The "Moof" dogcow logo has been on Apple's trademark list for at least three years. The dogcow logo appears on Apple's Developer CD Series and in other material. Apple also has a pending U.S. registration on it. Accordingly, it is not available to third-party developers as an icon or file symbol.

"Printer driver is MultiFinder compatible" bit

Written: 7/29/92

Last reviewed: 9/15/92

Could you please tell me what the "printer driver is MultiFinder compatible" bit is used for? We have been experiencing crashes related to resources and memory when this bit is off and have been experiencing crashes in the Color Picker when the bit is on. The "Learning To Drive" document is incomplete on this point.

The "printer driver is MultiFinder compatible" bit really provides two features. First, it allows the printer resource file to be opened by multiple clients. This was obviously needed to support multiple applications printing simultaneously under MultiFinder. The other feature provided by the flag is the loading of PDEFs into the system heap rather than the application heap (which is where they go under Finder).

The MultiFinder bit has a major limitation: If your driver has this flag set, you aren't allowed to add or size resources, or do anything else that would cause the RAM-resident resource map to change. Although MultiFinder lets multiple applications open the printer resource file at the same time, it has no control over the resource map that gets loaded by the Resource Manager when the file is opened. Because of this, each client gets its own personal copy of the resource map. When these guys get done with the file, they write the resource map back to the file (via UpdateResFile). Obviously, if the resources have changed in any way, the last guy to call UpdateResFile is the only one whose changes will be recorded. This is a "thrill seeker" method of handling the printing resource files, but since none of the Apple printer drivers currently add or resize resources, it made sense.

So the bottom line here is that if you want your drivers compatible under MultiFinder, you'll have to implement a scheme that doesn't require adding or resizing resources. It's OK to change the data in a resource, as long as you don't change its size. Changing the data will not cause changes to the resource map, so each client will still have duplicate copies of the map.

Here are two examples of what would happen to your printer driver's resources under Finder and MultiFinder when the "printer driver is MultiFinder compatible" bit is set:

- Under Finder with System 6.0.7: All resources are loaded into the application's heap—regardless of the resource attribute's bit setting. If the resource has the "load into the system

heap” bit set, it will still be loaded into the application’s heap under Finder. This makes sense in the Finder world because the application heap will usually have more room than the system heap.

- Under MultiFinder with System 7.0: All the printer driver’s resources **will** be loaded into the system heap. This is true whether the “load into the system” bit is set or not.

Why does the loading of your resources occur this way, even when the resource’s load into the system heap bit is set? Patches to `_GetResource` load all your printer driver’s resources into the system heap when the “printer driver is MultiFinder compatible” bit is set under MultiFinder,

and into the application's heap under Finder (as described above), which is why you can't override this behavior.

By the way, are you aware of the "SetPDiMC" MPW tool? It is available on the latest Developer CD Series disc. It will automatically set the bit for you when you build your printer driver. You might find this tool useful.