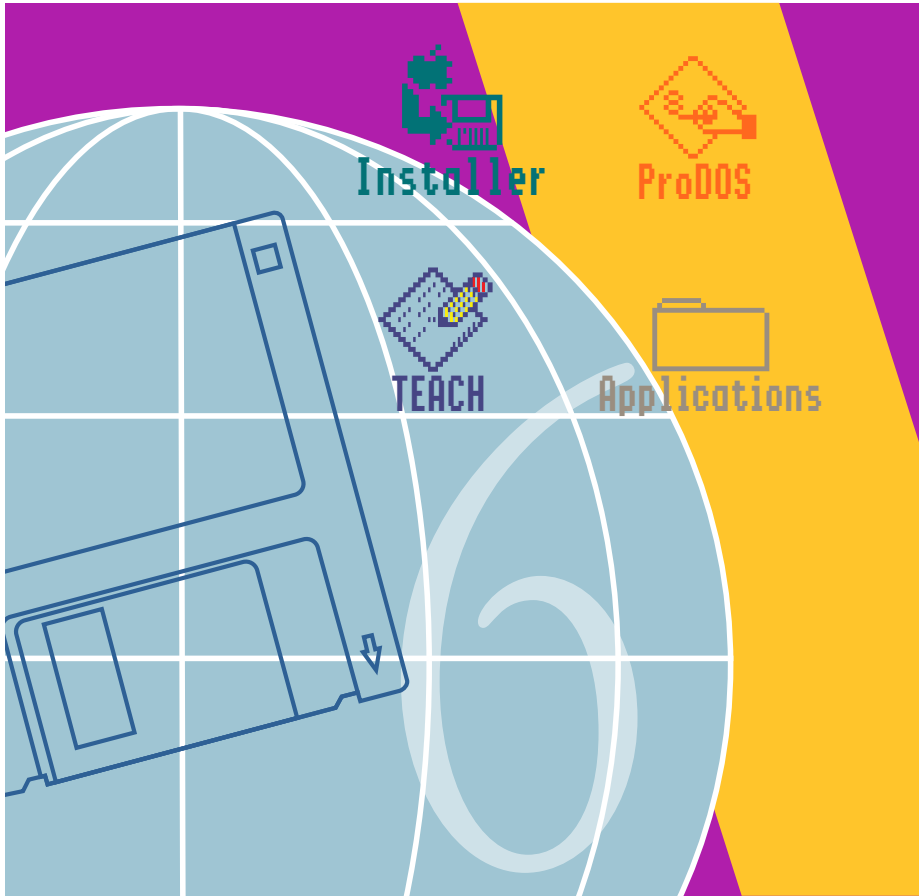




*StyleWriter Printer Driver
Delta Description*

Revision 1.3



System 6 IIGS

Apple® IIGs® System Software 6.0 — Release Notes
Golden Master 3 Release — March 5, 1992

STYLEWRITER DRIVER ERS

Apple IIGs StyleWriter Printer Driver

Delta Description

Revision 1.3: 9-October-1991

Ben Koning

PDF by khaibitgfx@outlook.com

September 6, 2019

Apple, the Apple logo, GS/OS®, and IIGs® are registered trademarks of Apple Computer, Inc.

© Apple Computer, Inc., 1991

20525 Mariani Avenue

Cupertino, CA 95014-6299

(408) 996-1010



khaibitgfx@outlook.com

Contents

Introduction, Scope, and References /	1
User Interface /	2
System Requirements /	3
Differences from other Printer Drivers and New Features /	4
No Draft Mode; No Color /	4
No Landscape /	4
No Reduction Modes; 80x36 DPI Coordinate Specification Only /	4
Improved Setup And Job Dialogs /	4
Status Message Disable /	4
Low-Memory Warning /	5
Error Handling /	5
Serial Communications Is Built Into The Driver /	5
StyleWriter Serial Communications Handshaking /	6
Interrupt Disable Time /	6
Immediate Mode Only /	6
New Handling Of Page-As-Job Application Behavior /	6
Color-To-Grayscale Mapping May Not Be Correct In All Cases /	7
Color-To-Grayscale Mapping Algorithm /	7
320-Mode Color-To-Grayscale Mapping Will Be Improved Soon /	7
640-Mode PixelMap Rendering Will Be Improved Soon /	7
Limited PrPixelMap Call Functionality /	7
Paper Dimensions /	8
Rasterization Scaling Of Application Graphics Objects /	8
Horizontal Line Layout For Text Objects /	8
Vertical Descender Clipping “Fix” /	9
Vertical Condense (“Height”) Operation /	9
Larger Banding Buffer Size Speed Improvement /	10
InstallWithStats Censoring Speed Improvement /	10
StdText Bottleproc TextBounds Censoring Speed Improvement /	10
Fast Host-To-Printer Pixel Conversion Speed Improvement /	10
Usage Of Full NikBits Commands Speed Improvement /	10
Optimized Compression Encoder Speed Improvement /	10
Selection Of Unidirectional Or Bidirectional Print Head Motion Modes /	11
Identification /	11

A Rough Idea Of Memory Requirements / 11
wDev Bit Summary / 12

Introduction, Scope, and References

This document briefly describes the technical and user aspects of the Apple IIGS printer driver for the Apple StyleWriter ink-jet printer. This driver allows 16-bit Apple IIGS toolbox (GS/OS) applications to use the StyleWriter.

The StyleWriter's physical operation is described in the *StyleWriter User's Manual* which is shipped with the printer. This manual is probably most important for its explanation of the printer's various paper feeding options.

The StyleWriter's usage from an engineering viewpoint is described in the *StyleWriter Printer Driver ERS*, written by Suki Lee. Embodied within this document are descriptions of the Macintosh StyleWriter printer driver as well as the full StyleWriter command set protocol.

Several important techniques regarding line layout are due to Steve Stephenson.

Usage of the Apple IIGS Print Manager (and therefore this and other printer drivers) is described in the *Apple IIGS Toolbox Reference*. Supporting information is found in Apple IIGS Technical Notes 34, 35, 36, and 46.

This is a feature-reduced printer driver, written in a very short time in order to get a driver completed as soon as possible. It is based on a similarly feature-reduced HP DeskJet PLUS driver which never shipped. Engineering emphasis has been concentrated around simplicity and performance rather than feature richness, since the StyleWriter demands a lot of host processing power and was not expressly designed with the low-horsepower Apple IIGS environment in mind.

User Interface

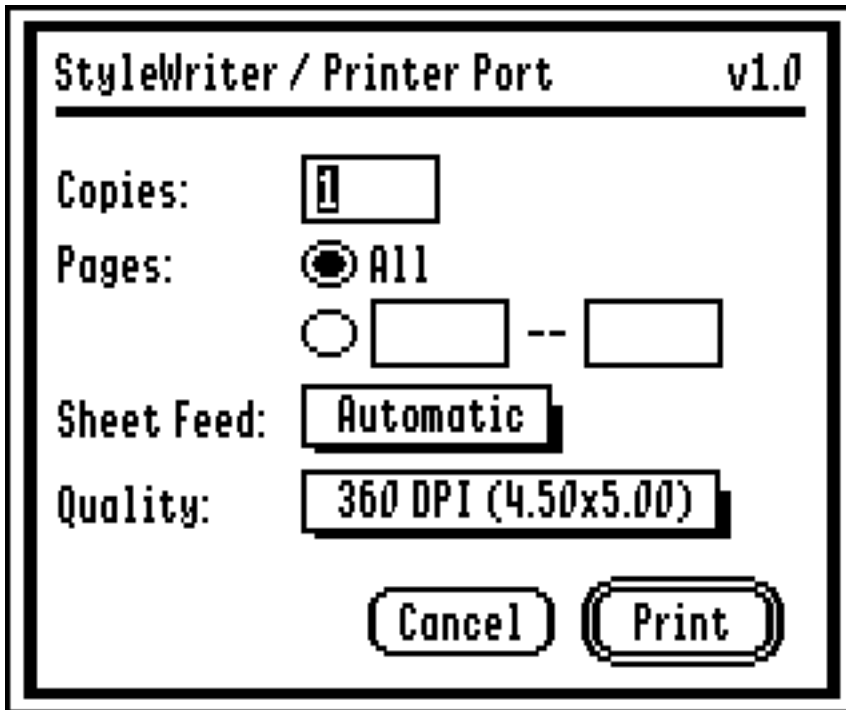


Figure 1. StyleWriter Print Job (PrJobDialog) Dialog Box

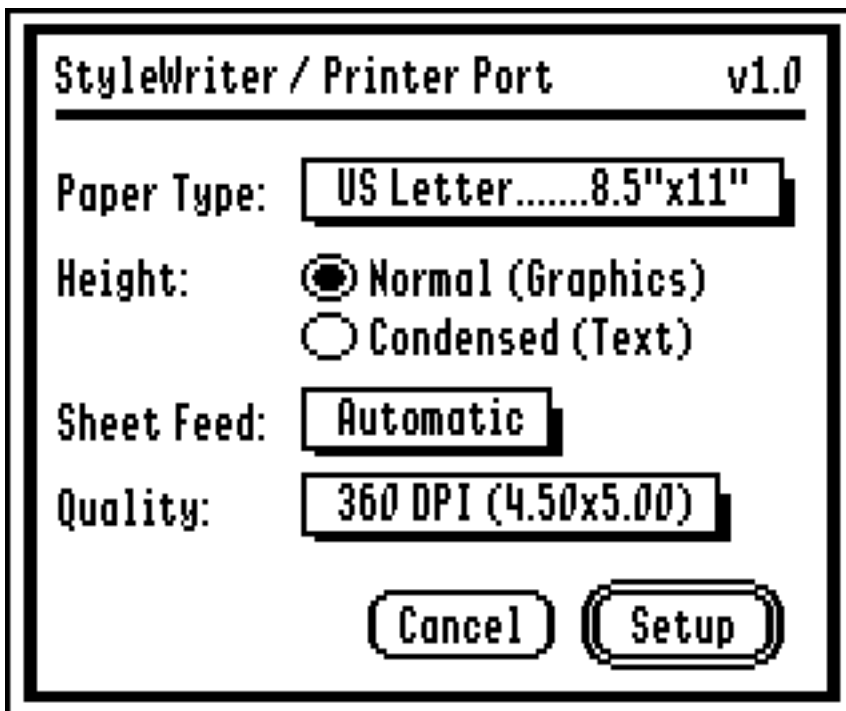


Figure 2. StyleWriter Page Setup (PrStlDialog) Dialog Box

System Requirements

The Apple IIGS driver is not shipped with the printer; however, the Macintosh driver is.

A mini-din serial printer cable is required to connect a StyleWriter to either Apple IIGS serial port.

The Apple IIGS StyleWriter Installer Script will install this driver, the PRINTER and MODEM print manager port drivers and the direct-connect printer CDEV.

If the user wishes to produce high-quality text output, he/she will also have to install large fonts using the appropriate installer script. The StyleWriter requires 5x fonts to produce high-quality output; if a font is not available, it will have to be scaled; this results in slower printing and low-quality output. As an example, if a string of text is entered into a word processor document as “Times 10”, then “Times 48” should be installed. Note that this is the reason for the extra large fonts disk in the Apple IIGS System 6.0 system disk set. (The Macintosh disk that is shipped with the printer includes the TrueType outline font rendering engine system extension (“INIT”) for pre-7.0 systems. TrueType allows the Macintosh driver access to large fonts without the necessity of shipping large font bitmaps on a separate disk.)

The drawing of 5x size fonts for printing places tremendous processing burden on the Font Manager scaling algorithm. A speed improvement will have to be done; at this time scaling a large font (from 96pt to, say, 82pt) takes about 40 seconds. The magnitude of this delay is unacceptable if it happens several (once for each font that appears in the document) or many (12 or more for each font under low memory conditions where fonts must be purged out of memory all time) times within a print job.

Two megabytes of memory and a hard disk with many large fonts are *highly* recommended. The driver's size on disk is about 15K; the entire suite of large fonts takes up an entire disk (about 700K).

Differences from other Printer Drivers and New Features

The end user and application writer will notice the following items to be different from previous printer drivers. All things not mentioned here should be the same as with other printer drivers.

No Draft Mode; No Color

There are no text-only “draft” or color modes; the printer is binary-value bitmap graphics output device only and does not support built-in fonts or color.

No Landscape

No landscape printing is possible with this driver at this time. Portrait (taller than wide) is supported only. Landscape printing means that the paper is still inserted the same way but the conceptual print area presented to the application (visible in rulers in word processors, etc) is rotated 90°.

No Reduction Modes; 80x36 DPI Coordinate Specification Only

No reduction modes are implemented (other than Vertical Condense). Reduction means smaller print in exchange for larger conceptual print area. As with all Apple IIGS printer drivers, there is currently no way by which an application may position the QuickDraw pen with printer-device resolution accuracy.

Improved Setup And Job Dialogs

The Page Setup and Print Job dialog boxes (Figure 1 and Figure 2) have been changed so that the user can not do page-setup-type things (such as altering the paper type setting) at job-dialog time anymore. However, the Print Quality and Sheet Feed selectors appear in both dialogs since they can be changed at both times. Code space is saved internally because the code has been integrated for both dialogs. If the version number is clicked, then an about-box alert is displayed.

Escape and Command-Period function as key equivalents for Cancel in addition to Return for OK. [Command]-H,h toggles the height setting (Normal/Vertical Condense); [Command]-M,m toggles the paper feed setting (Automatic/Manual).

Status Message Disable

The status box which is normally displayed at the top of the screen can be disabled by setting wDev bit 15 to “1”. This bit is always cleared by default and can be set only by an application. If this bit is set, the waitcursor is still shown/unshown, and the manual feed dialog still functions normally.

Low-Memory Warning

Under low-memory situations, the user may see a dialog box warning that fonts may be scaled and that thus they may not be positioned correctly within a document. This currently happens if less than 512K is returned by RealFreeMem and the user selected the (default) 360 dpi mode.

Error Handling

A warning dialog comes up if an application starts the printing loop after having set one or more unsupported features (such as by setting the Landscape orientation wDev bit). The dialog box comes up at PrOpenDoc time. The user can either cancel, which causes the print loop to abort with PrError=\$80, or continue anyway.

A dialog box now appears if the driver is unable to establish communication with the StyleWriter (ie, unable to obtain an expected response when sending the initial buffer space query or if there is an error in setting up the serial port (such as the slot being set to “your card”, etc)). This means that if the user has the slot mapped out, the printer turned off, not connected, etc., they will get this dialog in addition to the error message in the printer status box at the top of the screen.

Error handling for during printing has been added. A dialog box appears upon an error condition, which has the message text plus either one or two buttons, depending on whether the error is recoverable or not. There are three different messages which may appear: Out of paper, Ready light not on (ie, if in Manual Feed mode, you put in the sheet, and hit return to get out of the manual feed dialog while forgetting to press the Ready button first), and Serious problem has occurred. The Serious problem message happens in cases of pulling the serial cable, power cable, manhandling the carriage, etc; it specifically happens if any of the following printer controller error bits become asserted: Carriage Jam, Paper Jam, Bad Data Rectangle, Buffer Overflow, and Bad Self Test. The Out of paper and Ready light dialogs allow Cancel and Continue; the Serious problem dialog only allows termination of the print loop. The printer error status is checked whenever the printer buffer’s low-water mark is reached, so there may a few seconds of delay between an error condition and the error message. In addition, there is a new fail-safe no-printer-response timeout mechanism built into the communications code. This combination of error handling has proven to be fairly robust.

Serial Communications Is Built Into The Driver

Communications is handled internally inside the driver, due to the special way in which the StyleWriter’s 57.6K baud software-handshaking serial interface operates. The SCC channel A/B selection is inferred from the current printer-port driver settings: the PMGetPortName call is used; and a case-insensitive check is made for the first two characters of the “Printer” or “Modem” filenames returned by the PMGetPortName call. What this means to the user is that even though the internal communications data path is completely different as compared to other printer drivers, the user sees no difference in how they select one of the two output ports. The string that appears on the right-hand-side of the “StyleWriter / xxxx” title is shown as either “Printer Port” or “Modem Port”, rather than the actual filename of the currently selected port driver.

StyleWriter Serial Communications Handshaking

Since there is no hardware handshaking implemented in the StyleWriter, the driver's communications routines must ensure that the printer's buffer is not being overflowed with data. In order to assist with this, the printer is able to empty the buffer continuously while printing and while also accepting new input data, and the printer supports a command which causes it to return a byte indicating how much free space is left in its buffer. The StyleWriter is very picky about buffer overflows, since the incoming data is apparently edited and expanded by its controller in place. This driver is rather paranoid: It pins the incoming amount of free space query replies from the printer to 8192, and then sends data freely only as long as there are more than 4000 bytes in the printer's buffer. When the latter happens, the driver waits 4 seconds before attempting to gauge the amount of free space again, until there are at least 4000 bytes in the buffer. There is also a 5-tick (300mS) wait preceding any read attempt from the printer, so that the printer will have more than enough time to respond (the StyleWriter specification states that maximum response time for an immediate query reply should be 5mS).

Interrupt Disable Time

During transmission of a burst of data bytes to the printer, interrupts are disabled during most of the tight transmission loop: interrupts are disabled for 22 cycles on the average and enabled for 6 cycles. However, transmission does not happen most of the time; most of the time is actually spent rendering, imaging, and compressing, and during those activities interrupts are enabled.

Immediate Mode Only

Pages are recorded as pictures, rendered, and printed, one at a time; they are not saved up or spooled to mass storage as with other dumb-printer drivers. All copies of the current page are printed at PrClosePage time, like the LaserWriter driver. This has great simplicity and speed advantages but has the disadvantage that if a page image consists of many large pixelmaps, there may be low memory problems.

New Handling Of Page-As-Job Application Behavior

There are many programs (AWGS, HyperCardGS) which treat each page as a whole new document in order to save memory. This was a big problem with this driver because it must do an "immediate reset" printer command at the beginning of each document at PrOpenDoc time, which would immediately interrupt the preceding page, thus making it impossible to print multipage documents reliably. The implemented solution for this problem is that PrOpenDoc does not do the immediate reset (and other things such as not resetting the "Currently printing page: X" counter) ONLY IF the current tick count - last tick count saved by PrCloseDoc is less than an equated constant, which is currently set to 5 seconds. In other words, if a PrOpenDoc follows a PrCloseDoc within 5 seconds, then it is assumed that the page that is defined after the next PrOpenPage is part of the last job and not the start of a new one.

Color-To-Grayscale Mapping May Not Be Correct In All Cases

For speed-critical reasons, the standard white and standard black colors are assumed to exist in their standard color palette locations. If this is not the case then color to black/white mapping may be incorrect. This is not a problem unless the application sets an arbitrary color table entry to `color=$FFF` (white) and then uses it on the screen to represent white and assigns a nonwhite color to the standard color table entry for white (`%11(640)` or `%1111(320)`). The same goes for black (`%00` or `%0000`). We have not come across an application which does this on its own.

Color-To-Grayscale Mapping Algorithm

Color-to-grayscale mapping is as follows: Black screen pixels and white screen pixels are inverted to show up correctly as black or white on the printer. In 640-mode, both other possible screen pixel values are mapped to a horizontally-only alternating 50% duty-cycle dither pattern. Since most 640-mode graphics is dithered between the two colored pixel values and black or white, this actually results in five different gray shades. For 320 mode, the four-bit value is used to apply the 640-mapping algorithm twice, thus also resulting in five different, although blindly assigned, gray shades.

320-Mode Color-To-Grayscale Mapping Will Be Improved Soon

In the near future, the 320-mode printer-pixel generation algorithm (see immediately above) will be improved so that actual color values will be converted into halftone gray patterns. 640-Mode handling will remain the same.

640-Mode PixelMap Rendering Will Be Improved Soon

In the near future, the `StdPixels` routine, which is responsible for creating enlarged pixelmaps (in case any pixelmaps were drawn during the printing loop), will be patched out and improved for 640-mode, if not disabled, if memory is available, and if the Quality setting is set to 360 dpi. Currently, the `StdPixels` routine in `QuickDrawAux` handles 640-mode pixels a pair at a time, in order to preserve dithering relationships. This causes gray blurriness without any additional resolution around horizontal edges for applications which print 640-mode pixelmaps (such as `HyperCardGS`). The new `bottleproc` code will scale 640-mode pixelmaps a single pixel at a time. A new `wDev` bit, bit 12, will allow an application program to disable this process.

Limited `PrPixelMap` Call Functionality

The `PrPixelMap` call does not support printing of a subset of a pixelmap. Only the `LocInfo` structure is considered; the source rectangle is ignored. The printer connection is opened, the bitmap is printed using the 180 dpi mode, and the connection is closed before returning to the caller. Not very useful; but then an open connection/dump-bitmap/close connection architecture would need to be defined otherwise, however, since a bitmap describing a whole page would be prohibitively large.

Paper Dimensions

The following paper types are supported: US Letter, US Legal, #10 Envelope, and European A4 and European B5. 1/4 Inch margins are maintained on all four borders. The last two paper types are slightly different than the Macintosh driver which uses the Japanese A4 paper dimensions and doesn't support B5. The Apple IIGs driver is different because it follows the other Apple IIGs printer drivers. If the B5 or Envelope paper types are chosen, then the paper feed method is automatically set to Manual.

The following table gives the supported paper types and the resulting printable area (vertical dimensions are for Height=Normal):

- US Letter 8.0x10.5 inch 640x378 rPage (app drawable area) pixels
- US Legal 8.0x13.5 inch 640x486 rPage (app drawable area) pixels
- European A4 210x297 mm 625x405 rPage (app drawable area) pixels
- European B5 176x250 mm 530x337 rPage (app drawable area) pixels
- Envelope #10 9.0x3.625 inch 640x130 rPage (app drawable area) pixels

Rasterization Scaling Of Application Graphics Objects

The printer's resolutions, 180 and 360 dpi, are multiples (2.5 and 5) of 72 and not multiples of 80. This is why the Macintosh driver scales to 2.5x2.5 or 5x5 times the original graphic object size. Since Apple IIGs applications assume a 80x36 dpi page definition coordinate system, often without even checking the rPage/rPaper fields, the driver scales the recorded page definition picture to 4.5x (360 dpi) or to 2.25x (180 dpi) in the horizontal direction, and 5x (360 dpi) or 2.5x (180 dpi) in the vertical direction (the 360 dpi mode is default). Bitmaps are thus scaled; very rough patterns may exhibit some artifacts but almost always look acceptable. Graphic objects other than text (such as lines and ovals) are rendered correctly at full device resolution, since they are mathematically defined. Text strings must be scaled from available bitmapped fonts, however, and are scaled to 5x instead of 4.5x, to avoid the Font Manager speed and memory burden and total loss of visual fidelity which would result from scaling to the "strange" value of 4.5x. Note, however, that the positioning of these 5x text strings is still preformed at 4.5x.

Horizontal Line Layout For Text Objects

Normally, 4.5x positioning with 5x-size text font spacing would cause string overwriting and running off the right side of the page. However, the intercharacter spacing of text strings is manipulated by the print driver so that a given string played back during the page picture rendering process is drawn with 90% of its original width (note that 4.5 is 0.9 or 90% of 5). This is done by calculating small temporary negative values for SpaceExtra and CharExtra for each individual string within the patched-out StdText QuickDraw bottleproc. This general technique is also implemented (with more sophistication) in Macintosh drivers such as the Personal LaserWriter LS and is usually called "line layout". This behavior (which is always enabled by default) may be disabled by setting wDev bit 14 in the print record to "1".

The algorithm proceeds as follows:

1. The pixel width of a text string in the original point size which the application requested is re-discovered.
2. The original pixel width is multiplied by 4.5. This is the size that the string needs to be.
3. The desired width is compared to the actual width of the scaled font received from the Font Manager. This width is nearly always too wide, because the Font Manager scales the string width at 5x, not 4.5x.
4. The error delta is divided to be distributed among the spaces between words and all of the characters (including spaces). The spaces are given 1/7th of the responsibility.
5. The application's current CharExtra and SpaceExtra values are scaled up, in order to compensate for an apparent bug where QuickDrawAux does not scale them up. There are still bugs in this code as of this writing, however, and we're still not yet AWGS compatible.
6. The error delta (usually negative) is added to the scaled CharExtra and SpaceExtra values, and the string is drawn.

The string density is thus increased by 10% so that the application-assuming 80 dpi * 8 inch * 5 scaling-factor number of pixels is made to fit into the StyleWriter's 72 dpi * 8 inch * 5 scaling-factor number of pixels, and slight Font Manager scaling and QuickDrawAux face-modification (bold, italic, etc) errors are corrected so that horizontal overlap does not occur.

Vertical Descender Clipping “Fix”

In addition to horizontal error, vertical Font Manager scaling error also exists, so that a 5x font may actually be taller than 5 times the original font height (and we are scaling a true 5x in the *vertical* direction). The “fix” for this is to temporarily make the text transfer mode for all text strings ModeForeCopy. This substitution is made only if the original text transfer mode is ModeCopy (the default \$0000). A new wDev bit, bit 13, allows an application to disable this process. The code in currently released versions of the driver have a bug (they substituted ModeNotBIC instead, which causes gray boxes around colored text).

Vertical Condense (“Height”) Operation

The Height (Normal/Condensed) setting controls vertical spacing; in its default “off” state, every scan line is repeated at the printer. This doubles the vertical size so that 80x36 Apple IIGs bitmaps and nontext objects are WYSIWYG. But in this case, text fonts (which were all ported from the 72x72 Macintosh environment without any scaling) will appear twice too tall. When Vertical Condense is enabled, every scan line is not repeated at the printer, and the Macintosh fonts in the Apple IIGs are printed properly, with any surrounding Apple IIGs graphics vertically compressed. The Vertical Condense option also doubles the vertical conceptual printing area per page, as reflected by the rPage/rPaper rectangles in the print record.

Larger Banding Buffer Size Speed Improvement

Unlike older drivers, the banding buffer may expand beyond 64K if memory allows, thus causing fewer page image rasterization passes and thus enhancing throughput. However, it will never expand beyond 512K or 1/4 of the size returned by RealFreeMem, whichever is smaller.

InstallWithStats Censoring Speed Improvement

The Font Manager InstallWithStats call is patched out during page image rendering, in an attempt to prevent any unnecessary text scaling (which is very time-consuming). If the current pen vertical coordinate indicates that the piece of text whose font is being installed by InstallWithStats won't affect the band buffer, then we cause InstallWithStats to install the system font instead of the requested font. The system font requires no loading and scaling, and thus takes no significant amount of time to install. Note that this assumes that InstallWithStats is called after positioning the pen to the location of the text to be affected by the new font setting. The criterion used for censoring is that the current pen position must be at least 15 times the scaled-up requested point size number of pixels above or below the current banding buffer vertical limits.

StdText Bottleproc TextBounds Censoring Speed Improvement

The StdText patched-out QuickDraw bottleproc contains code that hides the text string that is about to be drawn from the rest of QuickDraw if that text string would not have any part inside the current rasterizing banding buffer, in order to speed up the rendering process.

Fast Host-To-Printer Pixel Conversion Speed Improvement

Apple IIgs color pixels are scanned in 16-bit words at once, optimized for scanning over white space; nonwhite words are converted immediately by means of a lookup table.

Usage Of Full NikBits Commands Speed Improvement

The repeat-scanline opcode is now being used for the non-vertical-condense case, and the all-blank-raster opcode is also being used if possible (in which case the compressor is bypassed altogether for that line). The compressor now uses the all-black-bytes and all-white-bytes NikBits opcodes. In previous versions of the driver, these features were not taken advantage of.

Optimized Compression Encoder Speed Improvement

The compressor has been optimized so that the rate at which it encodes large areas of same byte value is about twice as fast, and has been slightly optimized for the general case as well.

Selection Of Unidirectional Or Bidirectional Print Head Motion Modes

Bidirectional printing is automatically selected with 180 dpi mode; unidirectional printing is selected for use with 360 dpi mode. The choice is not explicitly given to the end-user.

Identification

The iDev print record field for this driver is 101 decimal (this is the same value as that used in the Macintosh driver); the initial version is \$0100 (v1.0). The PrChanged call calls PrDevPrChanged with a pointer to the three strings "=", "IJ10", "*", even though the printer cannot be networked.

IdleProc Temporal Granularity

The idleProc, if any is given in the print record, is called after processing every 128 raster lines. Command-period abort works even if an idleProc which doesn't check for command-period by itself is given.

A Rough Idea Of Memory Requirements

The process of printing, as implemented by this driver, requires memory to be allocated for the following purposes:

Driver	Memory Load
Print manager loaded	10K
Printer-port driver loaded	4K
Printer driver loaded	15K
Data structures/user interface memory	1K (grafports for windows, etc)
Banding/Rendering off-screen bitmap	Dynamic size, typical 75K, up to 512K
PICT definition of page contents	Minimum 8K, very much more if many bitmaps per page
Increased QuickDraw clip buffer sizes	$2.5 * \text{MaxWidth} \text{ SetBufDims parm } (=2K * \text{Max FontHeight})$
Increased FontManager strike/clip memory	$\text{Maximum FontHeight} * (\text{FontWidth} * 3 + 36 + \text{MaxWidth}) \text{ bytes}$
For each font brought into memory	53K typ (60pt font printing 12pt at 360dpi), ~130K max
For each pixelmap during PrClosePage	$\text{Pixelmap size} * 4.5$ temporarily during our patched StdPixels

wDev Bit Summary

The wDev bit field word in the print record for the StyleWriter (iDev=decimal 101) is currently defined as follows:

Bit Number	Name	Default Value	Description
0	wDevBetter	1	0=180 dpi, 1=360 dpi
1	wDevOrientation	1	0=Landscape (not supported), 1=Portrait
2	wDevVSize	1	0=Vert.Condense, 1=Normal (lines doubled)
3	wDevReduction	1	0=Reduction (not supported), 1=Normal
4		0	Unused
5	wDevColor	0	0=Grayscale, 1=Color (not supported)
6	wDevGap	1	0=No vertical gap (not supported), 1=Normal
7		0	Unused
8		0	Unused
9		0	Unused
10		0	Unused
11		0	Unused
12	wDevNo640PixMap	0	0=Normal, 1=Disable single-pix 640 pixmap scale
13	wDevNoForceTextMode	0	0=Normal, 1=Disable anti-descender-clipping fix
14	wDevNoLineLayout	0	0=Normal, 1=Disable horizontal text line layout
15	wDevNoStatusBox	0	0=Normal, 1=Disable print progress status box

If any unsupported feature is enabled (ie, by an application that chooses not to use PrStlDialog/PrJobDialog), then a dialog box with an option to cancel will be presented to the user at PrOpenDoc time.