Technote 1114

LaserWriter 8.5.1 Additional `CopyBits` Support: Transparent and Clipped Images

By David Gelphman, RBI Software Systems Inc.
and Ingrid Kelly, Apple Computer Inc.
devsupport@apple.com

CONTENTS

QuickDraw provides several APIs for copying bitmaps from one graphics port into another, the most flexible of which is `CopyBits()`. Most developers use `CopyBits()` because it can move offscreen images into an onscreen window, change colors in the pixmap, and resize images. This is a useful API, however, unfortunately, LaserWriter 8.4.x and earlier PostScript printer drivers from Apple have not supported all of the capabilities of the `CopyBits()` call to draw images. This Technote describes where Apple has enhanced its handling of the `CopyBits()` call in LaserWriter 8.5.1 so that developers and users may take advantage of the improvements.

CopyBits()

The prototype for `CopyBits()` is:

void `CopyBits`(const BitMap *`srcBits`, const BitMap *`dstBits`, const Rect *`srcRect`, const Rect *`dstRect`, short mode, `RgnHandle maskRgn`);

In the mode parameter, many developers have tried to use the transparent transfer mode to achieve transparency; the LaserWriter driver, however, never supported this mode but instead treated it as a `srcCopy` mode. The LaserWriter driver also has not supported clips in the `maskRgn` parameter. This has all changed with the LaserWriter 8.5.1 release.

Transparent Mode

One transfer mode previously unsupported by the LaserWriter driver is the transparent transfer mode. This mode allows drawing of images other than 1-bit deep so that any sample in the source which is the transparent color (defined as the QuickDraw background color) will not be painted to the destination. The advantage of this mode is that it lets a deep (> 1-bit) image have holes in it by coloring the holes in the image with the transparent color. Whereas previously the LaserWriter driver would paint these images as if the transfer mode were `srcCopy`, LaserWriter 8.5.1's new imaging code now handles the transparent mode.

maskRgn Clipping

Another previously unsupported feature of `CopyBits()` was the ability to pass in a mask (via the `maskRgn` field) to clip out portions of the destination image. Previous LaserWriter drivers completely ignored the `maskRgn` passed to `CopyBits()`. The new imaging code in LaserWriter 8.5.1 now supports the supplied `maskRgn`.

Note:
While the LaserWriter 8.5.1 driver now handles a supplied `maskRgn` to a `CopyBits` call, we do not do any special handling of the clip provided in the `grafPort`. `GrafPort` clipping is still handled by clipping only to the bounding rectangle of the port's `clipRgn`, both for `CopyBits` calls as well as line art and text drawing. Apple considered more generalized support for the port `clipRgn` but has rejected it to date mainly due to compatibility concerns.

Limitations

There are some known limitations with the added `CopyBits()` support that you should be aware of. They are:

- Extremely complex clipping regions may occasionally fail on Level 1 or Level 2 devices in low printer memory conditions. The result is that the output will not preserve the clip, and the image will print as if the `maskRgn` parameter is ignored.
  Specifically, for PostScript Level 1 printers, there is a hard coded limit to how complex a clip can be, regardless of the printer's installed memory. If the clip is more complex than this limit, the LaserWriter driver ignores the `maskRgn` parameter. For Level 2 printers, the limits are completely memory-based and generally much more flexible. However, once again, in low-memory conditions, the clip may still fail. On PostScript Level 3 printers, all clips should always print correctly regardless of the complexity of the clip since the driver uses the PostScript Level 3 masked image to support this feature.

- Since the LaserWriter driver implements the transparent mode via clipping, complex transparent regions may occasionally fail on Level 1 or Level 2 devices in low-memory conditions as described above. The result is that the output does not preserve the transparency, and the pixels image with the background color instead.

- For 2, 4, and 8-bit images with a color look-up table (CLUT), the LaserWriter 8.5.1 driver only supports one index which maps to the background color. Although the CLUT may contain many index values which have the background color and are therefore, in principle, transparent when displayed onscreen, the driver searches the look-up table for the first index which corresponds to the background color. Index samples with that index value are printed as transparent. Other index values which correspond to the background color are not treated as transparent but are printed as the background color.

- In order to maintain compatibility with many applications, the LaserWriter 8.5.1 driver will ignore a `maskRgn` which is an empty region under some circumstances (e.g. some rotated text and graphics). Some applications have used the fact that previous drivers always ignored the `maskRgn` parameter to clip out certain data when printing to QuickDraw printers. Relying on this 'feature' is not recommended, and the fact that there are some edge cases where a zero clip is ignored for compatibility reasons may be a temporary 'feature' of the driver. To avoid accidentally encountering these cases, when you really want a zero clip, either do not draw the image (preferred) or use `grafPort` clipping to clip it.

Summary

New imaging code in the LaserWriter has opened up some new functionality for the QuickDraw `CopyBits()` call. We encourage you to try LaserWriter 8.5.1 with your application and any PostScript device.

Further References

- Inside Macintosh: Imaging With QuickDraw, Chapter 3 and Appendix B.
- Apple's Technote web site
- PostScript Level 3 documentation from Adobe Systems Incorporated.

Acknowledgments

Thanks to John Blanchard, Tim Carroll and Paul Danbold.

Send feedback to devsupport@apple.com
Updated: 11-November-97

Tech Support
Technotes
Previous Technote | Next Technote | Contents

main section what's new 🍎 find it feedback help

Main | Page One | What's New | Apple Computer, Inc. | Find It | Contact Us | Help