

# Apple II Technical Notes



---

Developer Technical Support

## Apple IIGS

### #72: QuickDraw II Quirks

Revised by: C.K. Haun <TR> & Dave Lyons  
November 1990

Written by: Dave Lyons  
November 1989

This Technical Note points out some things things you need to watch out for when using QuickDraw II, especially with FastPort-aware and Shadowing modes.

**Changes since September 1990:** Added information about setting clipping buffer width with `SetBufDims`.

---

## Shadowing

Bit 15 of the `MasterSCB` parameter to `QDStartUp` requests that QuickDraw II use the memory at \$012000 to shadow onto the real screen memory at \$E12000. (This speeds up QuickDraw II, since memory in bank \$01 can be read at full speed.)

Previously, `QDShutDown` on ROM 01 machines was not disposing of the \$012000 memory block if it was allocated by `QDStartUp`. In 5.0.3, it's fixed—your application does not need to worry about disposing of the shadow memory manually.

Also, when `QDStartUp` uses shadowing on ROM 01 it does not actually clear the bank one memory to black—the screen looks black, but when you start to draw you get weird garbage around the stuff you draw (this is true in 5.0.3 and earlier). If you call `RefreshDesktop` before doing other drawing, it is not a problem. If you need to force the screen to clear, you can call `ClearScreen(0)` right after `QDStartUp`.

## **FastPort-Aware Anomaly**

When the FastPort-aware bit is turned on in the `MasterSCB` parameter to `QDStartUp`, `DrawPicture` does not notice changes in the pen pattern. If your application uses pictures, either directly or indirectly (i.e., by printing to the ImageWriter driver), you may need to leave FastPort-aware mode turned off to get the expected behavior.

## **FastFont and Large Pixel Maps**

FastFont does not work correctly when drawing past the first 64K of a pixel map. If you are drawing text that uses FastFont (i.e., Shaston 8), you can avoid this problem by using a non-rectangular `clipRgn`.

### ScrollRect Fixed in 5.0.3

`ScrollRect` no longer hangs when scrolling with a small positive `dY` and a non-rectangular `visRgn`.

### Don't ShowPen While Collecting Polygons, Regions, or Pictures

The Macintosh QuickDraw documentation permits calling `ShowPen` after an `OpenPoly`, `OpenRgn`, or `OpenPicture` call to cause drawing calls to contribute to a polygon, region, or picture **and** draw to a pixel map at the same time.

The Apple IIGS QuickDraw II documentation does **not** say you can do that. In some cases, it works, but it works “by accident” and it’s not one of the things Apple tests or guarantees in QuickDraw II.

### You May Need SetBufDims!

The call description for `SetBufDims` on page 16-215 of Volume 2 of the *Toolbox Reference* is misleading. The note in the description states, “You only need to make this call if your application is going to use, or allow the user to choose, fonts that have unusually large values of `chExtra` and `spExtra`.” This is **not** true; you need to call `SetBufDims` to adjust the clipping buffers for your application if you plan to use a `clipRgn` that has a greater width than the width you passed at `QDStartUp`.

`SetBufDims` sets the clipping buffer width as well as that of the text buffer, so if you plan to use a clipping region larger than the startup port width you must use `SetBufDims`.

Be aware that this call may be necessary even if your application does not ever set a clipping region or rectangle. Some toolbox calls assume that the clipping buffer size is correct based on the parameters passed to that routine. For example, if the `locInfo` you pass to `CopyPixels` has a `width` parameter that is wider than the width you passed at `QDStartUp`, `CopyPixels` may fail. A safe rule of thumb is to make sure (possibly by setting) that the `width` parameter in the buffer dimensions is the same or greater than the widest width in the `locInfo` structures passed to routines that use them.

### Further Reference

---

- *Apple IIGS Toolbox Reference*, Volumes 1 and 3