

# New Technical Notes

## Macintosh



---

Developer Support

### So Many Bitmaps, So Little Time

#### Imaging M.IM.BitMatToRegion

Revised by: Rich Collyer  
Written by: Rick Blair

December 1989  
April 1988

This Technical Note discusses the routine `BitMapToRegion`, which converts a bitmap to a region, and is available in the 32-Bit QuickDraw INIT and from Apple Software Licensing. **Changes since October 1989:** Added trap definitions for developers using the 32-Bit QuickDraw version of this routine without the correct MPW include file.

---

The following routine is now available to convert a bitmap to a region:

```
FUNCTION BitMapToRegion(region:RgnHandle; bMap:BitMap): OSErr;
```

in C:

```
pascal OSErr BitMapToRegion(RgnHandle region, BitMap bMap);
```

If you are using the 32-Bit QuickDraw version of this routine without the correct MPW include file, then you need to include one of the following definitions:

Pascal

```
FUNCTION BitMapToRegion (region: RgnHandle; bMap: BitMap): OSErr;  
    INLINE $A8D7;
```

C

```
pascal OSErr BitMapToRegion (RgnHandle region, const BitMap *bMap)  
    = 0xA8D7;
```

Assembly

```
_BitMapToRegion      OPWORD      $A8D7
```

The region will be built so that all one bits in `bMap` are inside the region and all zero bits are outside of it.

As with all QuickDraw calls which change a region, `BitMapToRegion` requires you to pass an existing region (originally created by `_NewRgn`). If the region cannot be built due to an insufficient heap space or a size greater than 32K, then the routine will return an appropriate error code and the region will be empty. If the region would have exceeded 32K, the error will be `rgnTooBigErr` (-500).

This function is useful for a number of situations where you have (or can produce) a bitmap representing an area. You can use `_CalcMask` to produce such a bitmap. Once you have a region, you can perform region operations (i.e., `_PtInRgn`, `_UnionRgn`, or `_InsetRgn`) or call `_DragGrayRgn`, for example.

This call is part of the 32-Bit QuickDraw INIT (\$A8D7). If you do not wish to depend on 32-Bit QuickDraw, then you can obtain a version of `BitMapToRegion` in MPW object format which can be linked into an MPW program, by contacting Apple Software Licensing:

Apple Software Licensing  
Apple Computer, Inc.,  
20525 Mariani Avenue, M/S 38-I  
Cupertino, CA, 95014  
(408) 974-4667  
AppleLink: SW.LICENSE

If you licensed the older version of this routine, `BitMapRgn`, contact Software Licensing about receiving an updated version. We recommend you update your application to use the new version as soon as possible.

The new version is now named `BitMapToRegion` to be consistent with the version in 32-Bit QuickDraw and the MPW interfaces. In addition, `BitMapToRegion` offers new features. You can now pass a one-bit pixelmap which has been coerced to a bitmap. If you pass a pixelmap which is too large, then you will get a `pixmapTooDeepErr` (-148) error. You can also pass the `portBits` of a window, much like you would do with a call to `_CopyBits`.

There is a potential problem with this routine, since MPW 3.1 include files contain information about 32-Bit QuickDraw. If you want `BitMapToRegion` to be available on all machines, then you must use the object file from Software Licensing. The problem is that when you compile your application with MPW 3.1 or later, the 32-Bit QuickDraw version gets preference over the object file. You **must** comment out the routine in the include files if you want to use the object file. If you only care about using `BitMapToRegion` on machines running 32-Bit QuickDraw, then you need not do anything.