ScalePt Resize coordinate pair to ratio of two rectangles

#include <Quickdraw.h> **Quickdraw**

```
void    ScalePt(thePoint, numerRect, denomRect);
Point   *thePoint ;     address of Point to scale; receives result
Rect    *numerRect ;    address of base Rect ('numerator" of ratio)
Rect    *denomRect ;    address of Rect to scale to ("numerator")
```

ScalePt converts each coordinate of a point to a value calculated as the ratio of the sizes of two rectangles. You could use this function to scale an object (such as a pen size or a character rectangle) to match the size of a smaller or larger window.

thePoint is the address of the point to convert. Upon return, its horizontal coordinates has been scaled by the ratio of the two rectangles' widths and the vertical coordinates has been scaled relative to the heights. It will never be set less than (1,1).

numerRect and . . .
denomRect are addresses of rectangles. Their location is irrelevant; their widths and heights are used to calculate ratios, which are applied to the coordinates of *thePoint* .

Returns: none

Notes: Since a point is an undimensional object, you can't really scale it. The **ScalePt** function treats *thePoint* as if it were the bottom-right corner of a rectangle whose top-left corner is at (0,0). That imaginary rectangle is scaled to the ratio of *numerRect* divided by *denomRect* . This call is functionally equivalent to:

```
numerWide = numerRect.right - numerRect.left;
denomWide = denomRect.right - denomRect.left;
numerHigh = numerRect.bottom - numerRect.top;
denomHigh = denomRect.bottom - denomRect.top;

thePoint.h = (thePoint.h * numerWide) / denomWide;
thePoint.v = (thePoint.v * numerHigh) / denomHigh;
```

Note: *thePoint* will never be set smaller than (1,1).

Use **MapPt** to perform the more common operation of finding the coordinates of a point inside one rectangle that corresponds to a similarly-located point within another.