ShieldCursor Hide cursor while moving (or while in) a rectangle

#include <ToolUtils.h> **Toolbox Utilities**

```
void      ShieldCursor(shieldRect, offsetPt );
Rect      *shieldRect ;    area in which to hide cursor
Point     offsetPt ;       offset from global coordinate system
```

Use this to keep the cursor from appearing when it is enclosed by a specified rectangle and/or to hide the cursor while it moves. Use **ShowCursor** to cancel the effect of this call.

shieldRect is the address of an 8-byte Rect structure, expressed in either local or global coordinates. It identifies the size and position of the rectangle in which the cursor will be shielded.

offsetPt identifies the difference between the coordinate system used for *shieldRect* and the global coordinates system.

If *shieldRect* is in global coordinates, use (0,0).

If *shieldRect* is in local coordinates, use the coordinates of the top-left corner of the current grafPort's portBits.bounds rectangle.

Returns: none

Notes: This function can be used to tell the mouse support to keep from modifying a particular section of the screen; e.g., while doing animation. The cursor is invisible while it is moving. When it stops moving, it will be drawn only if it is positioned outside of *shieldRect*. Use **ShowCursor** to balance this call and to cancel its effect. Use **HideCursor** for normal cursor hiding.

The *offsetPt* parameter exists because the mouse support works in global coordinates. Here's how to use this function if you need to use the local coordinate system for *shieldRect* :

```
Rect    shieldRect;

SetRect( &shieldRect, 10,10, 200, 200); /* local coordinates */
ShieldCursor( &shieldRect, thePort->portBits.bounds.topLeft );
    .
    :
ShowCursor(); /* when you no longer need to shield */
```

An alternative method is to convert *shieldRect* to global coordinates (via **LocalToGlobal**) before the call, and pass (0,0) as *offsetPt* .

THINK Reference © 1991-1992 Symantec Corporation