LUpdate Redraw list; handle update events

#include <Lists.h> **List Manager Package**

void	**LUpdate(***theRgn*, *theList* **);**	
RgnHandle	*theRgn* ;	handle leading to Region; the area to update
ListHandle	*theList* ;	handle leading to a ListRec

Call **LUpdate** in response to an update event for the list's window. It redraws the cells that intersect a specified region (usually the window's visRgn) and updates the scroll bars if needed. If drawing is off, this does not affect the list's visible area.

theRgn is a handle leading to a variable-length Region structure. It identifies the portion of the list you wish to update. Normal usage is to pass the window's entire visible region, e.g.:

```
LUpdate( (*theList)->port->visRgn, theList );
```

You can specify a smaller region (e.g., the rectangle of one or two cells) if you don't want to update everything (see **LRect** and **RectRgn**).

theList is a handle leading to a variable-length ListRec structure. It is a value previously obtained via **LNew**.

Returns: none

Notes: When an update event occurs for the window that encloses a list, call **LUpdate** to force the List Manager to redraw the parts of the list that need updating. For instance:

```
WindowPtr    listWindow;                    /* assumed to be created */

if(WaitNextEvent(everyEvent, &theEvent, 0, nil)) {/* in event loop */

    if ( theEvent.what == updateEvt ) {
        if ( theEvent.message == (long)listWindow ) {
            BeginUpdate( listWindow );
            LUpdate( listWindow->visRgn, theList );
            DrawGrowIcon( listWindow );           /* if needed */
            EndUpdate( listWindow )
        }
    }
}
```

Update events are generated when you call **InvalRect** for a portion of the list window or when another window gets moved, uncovering all or a portion of the list. See **GetNextEvent** and **BeginUpdate** for related information.

Hint: You may find it advantageous to store the ListHandle (*theList*) into its window's WindowRecord.refCon field; that way, you can use

GetWRefCon to get the value to use for *theList* in an **LUpdate** call from the event loop.

Remember, when drawing is off (see **LDoDraw**), this function has no effect on the list display.