LScroll

Scroll list by specific number of rows and columns

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

void	**LScroll(***deltaClms*, *deltaRows*, *theList* **);**	
<u>short</u>	*deltaClms* ;	distance to scroll right (>0) or left (<0)
<u>short</u>	*deltaRows* ;	distance to scroll down (>0) or up (<0)
<u>ListHandle</u>	*theList* ;	handle leading to a <u>ListRec</u>

LScroll scrolls (repositions) the list within its viewing area by a specified number of rows and columns. The scroll bars (if any) are updated. If drawing is on, the list is redisplayed immediately.

deltaClms is the horizontal distance to scroll the list, in columns. Values greater than 0 scroll the list toward the right (causing information to move toward the left), and values less than 0 cause the list to scroll left.

deltaRows is the vertical distance to scroll the list, in rows. Values greater than 0 scroll the list down (causing information to move upward), and values less than 0 cause the list to scroll up.

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

Returns: none

Notes: **LScroll** does not really "scroll" the data; it jumps it in the specified direction(s). It's an efficient operation since it appears to use the Quickdraw **<u>ScrollRect</u>** function to move the bulk of the image (invalidating the vacated area and redrawing only the cells that were previously hidden from view).

Over-large values for *deltaClms* and *deltaRows* are pinned to the data bounds; e.g., scrolling above row 0 will stop at row 0 (if <u>ListRec</u>.dataBounds.top is 0).

One case where you might use **LScroll** is to take action on a press of the PageUp or PageDown keys. Another case is when you want a particular cell to be positioned at the top of the list (e.g., a default selection). For instance:

```
SetPt( &theCell, 0,0 );                      /* search from top of list */
LSearch( "Geneva", 6, nil, &theCell, theList );   /* where's Geneva? */
LDoDraw(FALSE, theList );                    /* temporarily off */
LScroll( -1000,-1000, theList );             /* go to a reference point */
LDoDraw(TRUE, theList );                     /* back on */
LScroll( theCell.h,theCell.v, theList );     /* put Geneva on top */
LSetSelect( TRUE, theCell, theList );        /* pre-select it */
```

Note that **LScroll** uses relative positioning. The example first scrolls to the top of the list by scrolling up and left by 1000. The scroll will actually stop at 0,0. Next, it scrolls to the desired position, based on the cell coordinates.

See **LAutoScroll** for a way to scroll directly to the first selected item in a list.