BlockMove Copy memory from one place to another

#include <Memory.h> **Memory Manager**

void	**BlockMove(***srcPtr*, *destPtr*, *byteCount* **)**;	
Ptr	*srcPtr* ;	address of data to move
Ptr	*destPtr* ;	address to move it to
Size	*byteCount* ;	size of block to move, in bytes

BlockMove copies a block of memory from one place to another. It works correctly, even if the source and destination overlap.

srcPtr is the address of the beginning of the block to copy.

destPtr is the address at which to copy the block.

byteCount specifies the length of the block to copy. As a Size data type, it should be positive and less than 8MB (a 24-bit value).

Returns: none

Notes: This is a generalized block-move or block-copy operation that, for medium to large blocks, is much faster than writing a loop in C. Also, **BlockMove** can handle overlapping data areas.

For instance, the following example inserts one byte into the front of a 30-byte buffer:

```
char    theBuf[30];

BlockMove( &theBuf[0], &theBuf[1], 29 );
theBuf[0]='A';
```

Several higher-level tools exist for copying specific types of data:

HandToHand	create handle and copy handle data to it
PtrToHand	create handle and copy arbitrary data to it
PtrToXHand	copy arbitrary data to an existing handle
HandAndHand	concatenate handle data to another handle
PtrAndHand	concatenate arbitrary data to end of handle
SetString	copy handle data to a pascal-style string
PackBits	move and compress binary data
Munger	insert/replace data (usually text)
GetScrap	copy data from the scrap to a handle