HUnlock Unlock a handle's data (allowing it to be moved)

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

```
void      HUnlock(theHandle);
Handle    theHandle;      handle to be unlocked
```

HUnlock undoes the effect of **HLock**. It removes the relocation lock on a handle. In the event of a memory crunch, the Memory Manager will be able to move the handle's data to make room for other allocation blocks.

theHandle is a handle leading to a relocatable memory block. It is typically a value obtained from **NewHandle**.

Returns: none; the **MemError** function may return an Error Code of:

noErr	(0)	No error
nilHandleErr	(-109)	Illegal operation on an empty handle
memWZErr	(-111)	Illegal operation on a free block

Notes: To avoid heap fragmentation (i.e., to keep as much of the heap available as possible), use **HUnlock** as soon as possible after locking it via **HLock**. If you expect the handle to be locked for a long time, use **MoveHHi** to place it at the top of the heap, or use **ResrvMem** before allocating the handle to place it near the bottom of the heap.

If the block is already unlocked (its default state upon allocation), **HUnlock** does nothing.