**DebuggerUnlockMemory** Make part of the address space movable

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

**Debugger Support Under Virtual Memory**

```
OSErr       DebuggerUnlockMemory(address, count);
void        *address    is the start address of the range that is to be
                        unlocked in RAM
unsigned long  count    is the size in bytes of that range
            returns     Error Code; 0=no error
```

The **DebuggerUnlockMemory** function reverses the effects of **DebuggerLockMemory**.

*address* is the start address of the range that is to be unlocked in RAM

*count* is the size in bytes of that range

**Returns:** an operating system Error Code.

| | | |
|---|---|---|
| noErr | (0) | No error |
| paramErr | (-50) | Error in parameter list |
| notLockedErr | (-623) | Specified range of memory is not locked |

---

Notes: **DebuggerUnlockMemory** makes the portion of the address space starting with address and continuing for count bytes movable in physical memory and eligible for paging again. Unlocking is applied to whole pages of the virtual address space. Unlocked pages are marked as cacheable.