HPurge — Make a relocatable block purgeable

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

void **HPurge(***theHandle* **);**
Handle *theHandle* ; handle to block being made purgeable

HPurge sets the purge attribute of a relocatable memory block so that in a subsequent heap compaction, its data can be purged.

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: If *theHandle* is currently locked (see **HLock**), it will remain locked and **HPurge** will have no effect until the handle is unlocked via **HUnlock**. Thereafter, *theHandle* will purged by the next general purge.

After a purge, all handles marked as purgeable will point to NIL master pointers and their data area will be lost. Be very careful to check for such empty handles before accessing the data, e.g.:

```
Handle myHandle;

myHandle = NewHandle( 1000 );              /* allocate space */
HPurge( myHandle );                        /* allow purge */
     .
     :
if ( *myHandle == 0 ) {                    /* data has been purged */
     ReallocHandle( myHandle, 1000 );      /* get some storage */
     .
     .    /* regenerate lost data; load resource, etc. */
     .
}
HNoPurge( myHandle );                 /* don't allow purge now */
     .
     :
```

Use **HNoPurge** to undo the effect of this function (first be sure that the handle hasn't been purged!).