SetZone Select a heap zone as the "current zone"

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

```
void        SetZone( heapZone );
THz         heapZone ;      address of a 62-byte Zone structure
```

SetZone activates (makes current) a desired heap zone. Most Memory Manager functions operate on the current heap zone.

heapZone is the address of a 64-byte Zone structure. It is either the application zone (global variable ApplZone at 0x02AA), the system heap (global variable SysZone at 0x02A6) or a value used as the *startPtr* parameter in a previous call to **InitZone**.

Returns: none (call **MemError** to check for an error)

Notes: As an expedient alternative to **SetZone**, you can simply store a THz (pointer to a Zone structure) in the global variable TheZone (at 0x0118).

SetZone is needed by applications that maintain multiple heap zones, or in the rare case where you may want to allocate an object in the system heap.

You may use **SystemZone**, or **ApplicZone** (or access the global variables SysZone or ApplZone) to obtain a valid value for *heapZone* . You can use **HandleZone** or **PtrZone** to learn which zone owns a particular handle or pointer.

For instance, to allocate some data in the system heap (which is guaranteed to be there on a subsequent invocation of your application), you might use the following sequence:

```
SetZone( SystemZone() );          /*make system heap current*/
myHandle = (myType)NewHandle( sizeof( myType) );
SetZone( ApplicZone() );          /* application heap current */
(*myHandle)->myField = myValue;   /* store a value in system heap */
```

THINK Reference © 1991-1992 Symantec Corporation