Zone Page 1

Zone structure

#include < Memory.h >

| typedef struct <b>Zone</b> { |  | <u>Offset</u>   | <u>Description</u>   |
|------------------------------|--|---|--|
| bkLim ;                      | 4  | 0   | Block limit; addr of byte after last in zone   |
| purgePtr ;                   | 4  | 4   | (used internally)  |
| hFstFree;                    | 4  | 8   | First free master pointer  |
| zcbFree;                     | 4  | 12  | Number of free bytes   |
| gzProc ;                     | 4  | 16  | Address of grow zone proc ( <b>SetGrowZone</b> )   |
| moreMast;                    | 2  | 20  | Number of master ptrs to allocate at a time  |
| flags;                       | 2  | 22  | (used internally)  |
| cntRel;                      | 2  | 24  | (not used)   |
| maxRel;                      | 2  | 26  | (not used)   |
| cntNRel;                     | 2  | 28  | (not used)   |
| maxNRel;                     | 2  | 30  | (not used)   |
| cntEmpty;                    | 2  | 32  | (not used)   |
| cntHandles;                  | 2  | 34  | (not used)   |
| minCBFree;                   | 4  | 36  | (not used)   |
| purgeProc;                   | 4  | 40  | Address of purge warning procedure   |
| sparePtr ;                   | 4  | 44  | (used internally)  |
| allocPtr;                    | 4  | 48  | (used internally)  |
| heapData;                    | 2  | 52  | First byte of data in the zone   |
|                              | 54   |   | (size of this zone-prefix information)   |
|                              | bkLim;<br>purgePtr;<br>hFstFree;<br>zcbFree;<br>gzProc;<br>moreMast;<br>flags;<br>cntRel;<br>maxRel;<br>cntNRel;<br>maxNRel;<br>cntEmpty;<br>cntHandles;<br>minCBFree;<br>purgeProc;<br>sparePtr;<br>allocPtr; | bkLim; 4 purgePtr; 4 hFstFree; 4 zcbFree; 4 gzProc; 4 moreMast; 2 flags; 2 cntRel; 2 maxRel; 2 cntNRel; 2 cntEmpty; 2 cntEmpty; 2 cntHandles; 2 minCBFree; 4 purgeProc; 4 sparePtr; 4 allocPtr; 4 heapData; 2 | bkLim; 4 0 purgePtr; 4 4 hFstFree; 4 8 zcbFree; 4 12 gzProc; 4 16 moreMast; 2 20 flags; 2 22 cntRel; 2 24 maxRel; 2 26 cntNRel; 2 28 maxNRel; 2 30 cntEmpty; 2 32 cntHandles; 2 34 minCBFree; 4 36 purgeProc; 4 40 sparePtr; 4 44 allocPtr; 4 48 |

typedef Zone \*THz;

Notes: Several Memory manager calls require (or return) a THz (pointer to Zone), including:

| <u>GetZone</u> | <u>SystemZone</u> | <u>HandleZone</u> |
|----------------|-------------------|-------------------|
| SetZone        | <u>ApplicZone</u> | PtrZone           |

The global variable <u>TheZone</u> (at 0x0118) contains the address of the Zone structure at the start of the current heap zone.

The gzProc field is set when you call <u>InitZone</u> or <u>SetGrowZone</u>. Set it to NIL (0) if you don't care. This proc will be called when more space is needed, as explained in <u>SetGrowZone</u>.

The zcbFree field contains the total number of free bytes in the zone. If the zone is fragmented in the least, you won't be able allocate this much memory in one block.

You can install a custom "purge warning procedure" by storing its address in purgeProc. Note that purgeProc gets overwritten whenever you call **SetResPurge**(TRUE).