

New Technical Notes

Macintosh



Developer Support

Debugging With PurgeMem and CompactMem

Overview

Revised by:
Written by: Jim Friedlander

March 1988
October 1985

If you are having problems finding bugs like handles that aren't locked down when they should be, or resources that aren't there when they're supposed to be, there is a handy technique for forcing these problems to the surface. Every time through the main event loop call:

```
PurgeMem(MaxSize); {MaxSize = $800000}  
size:= CompactMem(MaxSize);
```

`PurgeMem` will purge all purgeable blocks and `CompactMem` will rearrange the heap, trying to find a contiguous free block of `MaxSize` bytes. Obviously, this will move things around quite a bit, so, if there are any unlocked handles that you have de-referenced, you will find out about them very quickly.

Don't be alarmed when you see the performance of your program deteriorate drastically —it's because lots of resources are being loaded and purged every time through the main event loop. You might want to have a debugging menu item that toggles between glacial and normal execution speeds.

Please be sure to remove these two lines from any code that you ship!! In fact, neither of these two calls should normally be made from your application. They tend to undo work that has been done by the Memory and Resource Managers.

Further Reference:

-
- The Memory Manager