

# HeapSpy DA

Version 1.0

©1991 by Jeff Bock



HeapSpy is a desk accessory for viewing the contents of active process zones (heaps) under Multifinder and System 7. The different types of blocks in the zone (free, nonrelocatable, purgeable, relocatable and locked) are displayed in distinctive patterns. Also displayed is some useful information about the zone.

## Installation

HeapSpy is a desk accessory and is installed in the usual way in System 6 or 7.

## HeapSpy Display

HeapSpy displays a dynamically-updated map of the zone, with low memory starting at the top of the display. The window is not resizable, but if the zone is larger than can be accommodated by the display, the scroll bar is activated and you can scroll through the zone. Different types of blocks are displayed in different patterns. The types shown are free, nonrelocatable (created by NewPtr) and relocatable, purgeable and locked (created by NewHandle). The size of the block is proportional to the length of the line in the window, with 1 pixel of line length = 4 bytes in the zone. There is no distinction between contiguous blocks of the same type. That is, a 100 pixels long segment in the Locked pattern could represent 100 contiguous 4 byte blocks, a single 400 byte block, or anything in between.

Additional information displayed in the window is the total size of the zone, in k bytes, and k bytes of free memory (as returned by the FreeMem function). There are also two indicators "F.I." and "H.U."

F.I., or Fragmentation Index is defined as

$$F.I. = (\text{BytesAvail} - \text{MaxBlock})/\text{BytesAvail}$$

where BytesAvail is the total of free and purgeable bytes in the zone and MaxBlock is the size of the largest contiguous block that could be obtained by purging the heap.

An F.I. of 0 indicates an unfragmented heap (that is, all available memory is contiguous). An F.I. approaching 1 indicates a highly fragmented heap, in which the available memory is broken into many small non-contiguous blocks.

"H.U.", or Heap Use, is simply the fraction of the memory in the zone being used.

$$H.U. = (\text{ZoneSize} - \text{FreeBytes})/\text{ZoneSize}$$

The last piece of information displayed is the (approximate) address at which the cursor is pointing when it's in the map section of the window. When the mouse is moved into the map area, the cursor changes to the '+' cursor. HeapSpy displays the address of the block at the center of the '+'.

The display is updated periodically, even while HeapSpy is in the background. When HeapSpy is in the foreground, a spinning watch cursor is displayed while HeapSpy updates the zone map.

## HeapSpy Menu Commands

HeapSpy installs its own menu. The commands are:

### *Next Zone*

Display the zone map of the next process in Multifinder or System 7's process list. When selected, HeapSpy displays a spinning watch cursor while it parses the process's zone. It then updates the displays in the window. If the zone

can't be displayed because of insufficient memory (see *Limitations* below) or a bad block in the zone, HeapSpy puts up an alert and then tries to display the next process.

#### *Update Interval*

The HeapSpy display is updated at intervals selected from the "Update Interval" submenu. The frequency can be 2 to 64 seconds (the default is 8 seconds). Auto updating can be turned off by selecting the "Off" submenu item.

#### *Manual Update*

Selecting the "Manual Update" item causes HeapSpy to update the display immediately.

#### *Purge*

Purge purges the zone of the process being displayed by calling `_PurgeMem`. All memory occupied by relocatable blocks marked purgeable is released.

#### *Compact*

Compact compacts the zone of the process being displayed by calling `_CompactMem`. All relocatable blocks are moved as low in the zone as possible.

#### *Bring to Front*

Bring To Front makes the process being displayed the front-most process. By displaying an application's zone in the HeapSpy window and bringing the process to the front, you can watch memory being allocated and deallocated and moved around as HeapSpy dynamically updates its display.

#### *About HeapSpy...*

This item is available in System 6 only. It displays a dialog with the usual copyright and version information. The "About..." box is available in the Apple menu in System 7.

### **Uses for HeapSpy**

HeapSpy displays no information that can't be obtained with a low level debugger like TMON Pro<sup>®</sup> or MacsBug. However, for me at least, it displays the information in a more comprehensible way. I use HeapSpy to check for heap fragmentation and to estimate the amount of memory I need to allocate in an application's SIZE resource. With the block address feature, I can get the address of a locked or nonrelocatable block that is fragmenting the zone, enter TMON, and (usually) figure out what the block is and how it got there.

### **Limitations**

HeapSpy requires System 6.0.3 or later with Multifinder (either the release or the 6.1b9 version) running, or System 7, in 24-bit mode. It doesn't understand 32-bit heaps (neither does the programmer). HeapSpy can require a substantial amount of memory. It allocates an offscreen bitmap into which it draws the zone map. The bitmap requires about 1 byte for 8 bytes in the zone, so for MPW, with a partition of 2000k, HeapSpy allocates a 256k bitmap. If HeapSpy can't allocate the bitmap for a process, it displays an error message and tries to display the next process.

### **Source Code**

The (heavily commented ) MPW Assembly language source code for HeapSpy is available by sending a SASE with about \$.75 postage, (cheaper than downloading) and blank disk (or better yet a disk with some of your source code or other interesting stuff) to the author at

6418 Chippendale Dr.  
Baton Rouge, LA 70817

### **Disclaimer, Bug Reports**

I have conscientiously written HeapSpy and believe it to be reasonably stable. However, it's not been extensively tested. Also, users find many more bugs than programmers. In short, use it at your own risk. If you find a bug in HeapSpy please report it to me as soon as possible with as much information as possible. I would also appreciate comments and suggestions.

I can be reached on GENie (LINDSAY), America Online (JeffB27) or Compuserve (73260,1174).

### **Copyright, Distribution**

Unaltered copies of HeapSpy may be freely distributed if accompanied by this document. There is no charge for using HeapSpy. However, it is not in the public domain and I retain all rights.