Memory Management

The Macintosh Operating System manages the loading of applications, desk accessories, and other code into and out of memory. Applications must share the amount of memory available. Without virtual memory, if an application needs a greater amount of memory than is currently free for application use in the user's system, the user must free up some memory. With virtual memory, the Operating System can store elsewhere the contents of memory in use by other applications in order to make room for the active application.

Virtual memory extends the available memory beyond the limits of physical RAM by using part of the available secondary storage (such as a hard disk) to hold portions of programs and data not currently in use. When an application needs portions of memory stored on disk, the Operating System brings those portions back into physical memory by swapping them with other unused portions of memory.

The operation of virtual memory is mostly transparent to your application. The user sets options in the Memory control panel to control various features of virtual memory. The user chooses whether virtual memory is turned on and, if so, how much virtual memory is available. The main benefit of virtual memory is that it allows users to run more applications at once and work with larger amounts of data.

See **Memory Manager** for further information on using virtual memory.

Temporary Memory

Your application can allocate temporary memory if it needs additional memory for short-term purposes. Your application is not always guaranteed the desired amount of memory, so it should work correctly even if it does not get the requested memory. For example, you might allocate a small buffer in your application heap to copy data, and request additional temporary memory. If the temporary memory is available, your application can use it to copy large amounts of data more quickly. If the temporary memory is not available, your application should still be able to perform the copy, although it might take a little longer. As soon as your application finishes using the temporary memory, you should release it so that the memory can be made available to other applications.

See the **Memory Manager** for further information on using temporary memory.

24-Bit and 32-Bit Addressing

For Macintosh computers that support 32-bit addressing, the **Memory Manager** in system 7.0 uses all 32 bits of a memory address when the 32-bit addressing setting in the Memory control panel is on. Earlier versions of system software use 24-bit addressing, in which only the first 24 bits of a memory address are significant, and the upper 8 bits are ignored. For compatibility, all machines that support 32-bit addressing also support 24-bit addressing.

Macintosh computers that support 32-bit addressing can run with either 32-bit addressing or 24-bit addressing, but not both at the same time. The user chooses 32-bit addressing or 24-bit addressing by changing the setting in the Memory control panel and restarting the computer.

Applications that use the upper 8 bits of a memory address do not work correctly in 32-bit addressing mode. Applications that strip the upper 8 bits of a memory address or rely on the structure of the **Memory Manager** heap also do not work correctly in 32-bit addressing mode. Therefore, your application should not directly manipulate the bits in a memory address. If your application can operate correctly in 32-bit addressing mode, you can indicate this to the Operating System by setting a flag in your application's 'SIZE' resource. See the **Event Manager** for a discussion of the 'SIZE' resource.

If you use your own customized window definition functions or customized control definition functions, see the **Memory Manager** for guidelines on avoiding memory address violations. The **Memory Manager** also provides further guidelines on how to write an application that works with 32-bit addressing.