

**#213: `_StripAddress`: The Untold Story**

See also: *Inside Macintosh*, Volume V, The OS Utilities
Technical Note #212: The Joy of Being 32-Bit Clean

Written by: Andrew Shebanow October 1, 1988

Inside Macintosh, Volume V, The OS Utilities, incorrectly documents the `_StripAddress` trap; this Technical Note correctly documents the trap.

```
FUNCTION StripAddress(theAddress: Ptr): Ptr;
```

The `_StripAddress` trap now takes a `Ptr` as a parameter and returns a `Ptr` instead of a `LONGINT` as documented in *Inside Macintosh*.

Apple provides the `_StripAddress` trap to help developers ensure the legality of handle addresses. Because the current version of the Memory Manager stores its flags in the high byte of the master pointer, it is necessary to call `_StripAddress` before using a handle when in 32-bit mode. A call to `_StripAddress` might be appropriate in a driver for a NuBus card that is configured with more than one megabyte of memory. If the driver needs to copy data from a handle to the memory on the card, it needs to switch the Macintosh into 32-bit mode before it performs the copy. If the driver dereferences the handle without calling `_StripAddress`, it may be invalid, and you might get a bus error.

The true behavior of `_StripAddress` on the passed address is:

Operating System	24-Bit Mode	32-Bit Mode
Current Macs	masked with <code>Lo3Bytes</code>	masked with <code>Lo3Bytes</code>
A/UX	(never in 24 bit mode)	address unchanged
Future Macs	(never in 24 bit mode)	address unchanged

The `_StripAddress` trap is incorrectly described in the OS Utilities chapter of *Inside Macintosh*, Volume V:

```
FUNCTION StripAddress(theAddress: LONGINT): LONGINT;
```

If the system is running in 24-bit mode, StripAddress is identical in function to the global variable Lo3Bytes: it returns the value of the low-order three bytes of the address passed in theAddress. If the system is in 32-bit mode, however, StripAddress simply passes back the address unchanged.