

New Technical Notes

Macintosh



Developer Support

Use Care When Swapping MMU Mode Memory M.ME.SwapMMUMode

Revised by: Andrew Shebanow

February 1990

Written by: Cameron Birse

April 1989

This Technical Note describes how to avoid crashing when swapping into 32-bit mode on a Macintosh II. Thanks to Jim Berry and Dan Weston for pointing this out.

Changes since April 1989: Added a reference to Technical Note #213, `_StripAddress`: The Untold Story.

There is a condition where calling `_SwapMMUMode` to switch the Macintosh II into 32-bit mode can cause the system to crash. This condition happens in code which is loaded into memory from a resource, or is placed in memory that was allocated by the Memory Manager and is subsequently executed by using the master pointer as the address for a JSR instruction. This condition includes stand-alone, executable code resources (i.e., 'XCMD', 'XFCN', 'INIT', 'ADBS', 'FKEY', etc.), but does **not** apply to standard 'CODE' resources because the Segment Loader fixes the PC.

When you load code into memory as a resource in 24-bit mode (i.e., by calling `_GetResource`), the high byte of the master pointer contains Memory Manager information. If you perform a JSR to the code (typically a JSR (A0) with the master pointer in A0), the entire master pointer gets translated directly into the program counter, including the high byte of Memory Manager information. As soon as you switch into 32-bit mode, the program counter effectively has garbage in the high byte, and the machine goes directly into the weeds (do not pass go, do not collect \$200).

You can avoid this problem by cleaning up the program counter from within the resource code before calling `_SwapMMUMode`. The following example shows how to clean up the PC using MPW Pascal and C with inline assembly code:

MPW Pascal

```
PROCEDURE FixPC;
  INLINE  $41FA, $000A, { LEA      *+$000C,A0      }
           $2008,      { MOVE.L   A0,D0           }
           $A055,      { _StripAddress           }
           $2040,      { MOVEA.L  D0,A0           }
           $4ED0;      { JMP      (A0) ;jumps to next instruction }
```

MPW C

```
pascal void FixPC()
= {0x41FA, 0x000A, /* LEA      *+$000C,A0      */
  0x2008, /* MOVE.L   A0,D0      */
  0xA055, /* _StripAddress */
  0x2040, /* MOVEA.L  D0,A0      */
  0x4ED0}; /* JMP      (A0) ;jumps to next instruction */
```

Further Reference:

- *Inside Macintosh*, Volume V-591, OS Utilities
- Technical Note M.ME.32BitClean —
The Joy of Being 32-Bit Clean
- Technical Note M.ME.StripAddress —
_StripAddress: The Untold Story