

# New Technical Notes

Macintosh



---

Developer Support

## Register A5 Within GrowZone Functions

Memory M.ME.GrowZoneA5

Revised by:

March 1988

Written by: Chris Derossi

July 1987

---

If you have a grow zone function, it may get called when a system routine is trying to allocate memory. Because this can happen, you can't be guaranteed that register A5 will be correct.

If your grow zone function depends on A5, you should save register A5, load A5 from the low-memory global `CurrentA5` (a long word at \$904), and restore the caller's A5 before you exit.

From high-level languages, you can also use the Operating System Utility calls `SetUpA5` and `RestoreA5` (page 386 of *Inside Macintosh Volume II*). `SetUpA5` stores the 'old' A5 on the stack and puts the value stored at `CurrentA5` into A5. Make sure to call `RestoreA5` when you're done so that it can pop the saved value of A5 off the stack.

Your grow zone function depends on A5 if it does any of the following:

- Accesses your application's global variables (which are stored at negative offsets from A5).
- Accesses the QuickDraw globals. (A5 contains the address of a pointer to the QuickDraw global variables.)
- Makes any ROM trap calls.
- Makes any intersegment calls to routines in your application.

To do any of these, A5 needs to contain the value from `CurrentA5`. Please note that this is different than the method for calling the ROM from trap patches, where A5 should retain the value it had upon entry to your patch.

**Further Reference:**

---

- The Memory Manager
- Technical Note M.ME.GrowZoneA5 —  
Register A5 Within Trap Patches