



# Tech Info Library

## Applesoft: OUT OF MEMORY errors

Revised: 11/7/84  
Security: Everyone

Applesoft: OUT OF MEMORY errors

=====

There are only two ways to run out of memory in Applesoft. The most common reason for this error is either that your program is too large or it's using too many variables. The solution: trim down your program, keep the data on a disk, or chain your program in from the disk in segments.

The less common cause of this error is stack overflow. This problem is easily spotted, since after you receive the OUT OF MEMORY error, PRINT FRE(0) will tell you that there is still free memory. The 6502 stack, though, which Applesoft uses to save its recursive subroutine calls, is a limited resource; it's this stack that your program is overloading. Your program is overloading this stack in one or more of the following ways:

1. Too many FOR-NEXT loops
2. Too many GOSUBs
3. Excessively complex mathematical or string functions
4. GOSUBs with no RETURN
5. Improper recovery in ONERR GOTO routines
6. CALLs or interrupts that don't restore the stack properly

The effects of these functions are cumulative, so a combination of them could be causing your error.

Items one through four commonly occur in very complex programs. If one or more of these items is the cause of your problem, you can implement a program-controlled restart by executing the Applesoft routine CALL 54915; this routine clears the stack without clearing the variables by eliminating all pending FOR-NEXT loops, GOSUBs and formulas. Items one through four, though, have to do with your program structure, so when your program is cleanly structured you shouldn't have problems overloading the 6502 stack.

If a stack overflow is caused by running a cleanly-structured program, then that implies that the overflow is being caused by items five and/or six.

If you're using ONERR GOTO, carefully read pages 81, 82 and 136 of the (green and white) Applesoft BASIC Programming Reference Manual. There are two correct ways to leave the ONERR GOTO routine: (1) use a RESUME so Applesoft can take

care of the stack and re-execute the statement that caused the error, or (2) use the stack recovery routine on page 82 or 136 before any GOTO statement. (The example on page 136 is easier to implement.) If you're using Applesoft in ROM, replace the routine on page 136 with CALL -3288 and no POKEs.

When Applesoft executes a CALL, it does a 6502 JSR to the specified address. Other routines, such as those that handle interrupts from a peripheral card, also involve the 6502 stack. It's up to you to leave the stack the way you found it--so if you use any of these routines, make sure that you remove any bytes the pushed onto the stack so that the 6502's internal registers are maintained.

Apple Tech Notes

Tech Info Library Article Number:65