



# Tech Info Library

## Pinpoint: BASIC Programming Considerations

Revised: 1/20/86  
Security: Everyone

Pinpoint: BASIC Programming Considerations

=====

### Memory Usage

When Pinpoint is installed to operate in conjunction with the ProDOS interpreter BASIC.SYSTEM, it occupies some memory space on the \$300 page. It requires the memory space from \$300 thru \$3BA. This area is not used at all by BASIC, but is often used by small machine language programs. If a small machine language program is loaded here, you will lose control of your machine keyboard.

In addition, a 16K portion of the /RAM drive is reserved for Pinpoint, reducing the block count available on the RAM drive by 32 blocks. The Pinpoint Dispatcher is stored away in the upper 16K of the Extended memory bank.

### Key trapping

If your program uses the standard BASIC character reading capability (i.e. Input, Get), then Pinpoint can trap the Solid-Apple-P key. If you call a machine language routine for keyboard input, it will not be intercepted by Pinpoint. If you wish to check the \$C000 strobe in your machine language program, you can call \$32D. It will return the standard result of an LDA \$C000 providing that a Solid-Apple-P has not been hit.

### Invoking Pinpoint directly from BASIC

If you wish, you can invoke Pinpoint directly from your program. This would allow you to re-assign the Pinpoint key. Simply CALL 883 from BASIC (JSR \$373 from Machine Language). This will cause the Pinpoint Accessory Menu to appear.

NOTE: Apple Computer Inc. is not responsible for the contents of this article.

Pinpoint Publishing

Tech Info Library Article Number:1662