

RAM MEMORY MANAGER (CALL PRAM)

New way to use RXB way ahead of any other XB made is PRAM that allows you to change the size of RAM in upper 24K of RAM. Normally >A040 is the end of RAM in XB as it starts going from high RAM >FFFC down to lowest toward >A040 this allows 64 bytes not used but was for the TI Debugger to use. The PRAM command changes the location of the end of XB RAM. Normally XB RAM is >A040 in hex so the PRAM command allows changing this location to as low as 298 bytes of XB RAM. Any location from >A000 to >FEBE is a valid change in PRAM. Thus -322 decimal or >FEBE hex is highest address is -25576 decimal or >A000 hex lowest address. That tops our XB RAM to 64 more bytes then normal at max or down to 298 bytes of RAM. How come no one else thought of this? (Need to fix program start)

VDP STACK MEMORY MANAGER (CALL VDPSTACK)

Normal VDP stack location is 2392 in decimal >0958 in Hex. Some XB programs like The Missing Link use 6176 or >1820 Hex. Another location would be like 4096 which is >1000 in Hex. The VDPSTACK subprogram allows change of location of the VDP STACK in VDP RAM. Care must be taken to where you place the stack after all any over write or change can crash XB. Changing the VDP stack location allows changes in type of VDP mode being used like TEXT mode or Multi colored mode.

FILES BUFEEER MEMORY MANAGER (CALL FILES)

The FILES subprogram differs from the Disk Controller FILES on the CorComp, TI, Myarc or Parcom versions. All of these require a NEW after CALL FILES. NEW is executed after the FILES subprogram in RXB, no need to use NEW it is built into FILES. Also RXB FILES accepts values from 0 to 15 unlike the other FILES routines that can only accept 1 to 9. Each open file reduces VDP by 534 bytes, plus each file opened will use 518 bytes more. CALL FILES(0) will display 5624 Bytes of Stack free and 24488 Bytes of Program space free. At this point up to 15 files may be open at the same time. Not recommended but possible. Thus RXB 0 files now is possible in RXB or up to 15.