RXB TO ASSEMBLY DIRECT ACCESS BY ADDRESS:

  EXECUTE is much faster than the traditional LINK routine built
into XB. The main problem with LINK is it checks everything and
pushes everything onto the VDP stack. After getting to Assembly
it pops everything off the stack for use or pushes what is to
be passed to XB onto the stack. EXECUTE on the other hand just
passes a address to a 12 byte Assembly program in Fast RAM and
RTWP ends the users program. A LINK will use up 6 bytes for the
name, 2 bytes for the address and wastes time checking things.
  The advantage to EXECUTE is you use LOAD or MOVE or MOVES to
place the values needed directly into the registers then do it.
EXECUTE uses less space, is faster, and is easy to debug.

SAMS SUPPORT ROUTINES:

  The SAMS has support routines built into RXB. CALL SAMS("MAP")
will turn the SAMS mapper on. CALL AMS("PASS") turns SAMS mapper
to pass mode. CALL SAMS("ON") will turn on the read/write
lines of the mapper. CALL SAMS("OFF") turns off the read/write
lines. With these commands pages of memory can be written with
a CALL LOAD or read with a CALL PEEK.
  RXB AMS SUPPORT USES NO ASSEMBLY OR CALL LINKs. That means up
to 1 meg of 4K pages in entire 32K from RXB. That is impossible
to do from XB as you have to load your normal support somewhere
in 32K of assembly for everyone else not using RXB.
  GPL is where all the support routines are stored in RXB so not
one byte is wasted on assembly support. That also means not one
byte of SAMS memory in wasted on control routines.
  Speaking of control CALL SAMS switches 4K pages in the 32K SAMS.
CALL SAMS uses boundry symbols upper case only.
i.e. 2 = >2000, 3 = >3000, A = >A000, B = >B000, C = >C000,
D = >D000, E = >E000 and F = >F000

RND FUNCTION REPLACED

Extended Basic RND has been replaced with the TI BASIC RND as the
normal XB version of RND was hindered by to much Floating Point
that is very slow for use just to get a random number. Also the
XB RND was insanely complicated and bloated.