

RADIUS SANE TECH NOTE
(version 1.2)
6/9/88

WHAT IS IT?

Radius SANE is a floating point arithmetic package designed for speed. It works with any Macintosh Plus or SE that is equipped with any type of Radius Accelerator and a MC68881 floating point co-processor (or the Macintosh II). Since it is based on Apple's Standard Apple Numeric Environment (SANE), it is compatible with existing applications and will make any program that normally uses Apple SANE for its floating point arithmetic run faster.

Speed improvements are quite remarkable. For instance, trigonometric calculations in Radius SANE are approximately 15 times faster than Apple's MC68881 version of SANE (found on the Macintosh II), and approximately 50 times faster than Apple's non-881 version of SANE (found in the SE or Plus). Arithmetic calculations are approximately 1.5 and 3.5 times faster, respectively.

HOW DO I USE IT?

Using Radius SANE is as simple as copying the "Radius SANE" file into your System Folder and rebooting. The Radius SANE file is an INIT which installs itself into memory during boot. It works at the system level by intercepting all calls that normally go to Apple SANE. Thus, any application that uses standard SANE calls will now go through Radius SANE instead. Removing Radius SANE is just as easy — simply take the "Radius SANE" file out of your System Folder and reboot.

WHAT SACRIFICES WERE MADE TO ACHIEVE SPEED?

Very few, actually. To begin with, SANE is an incredibly well-defined and precise floating point environment. It is so precise, in fact, that it exceeds the IEEE (Institute of Electrical and Electronics Engineers) standards for floating point operations. In writing their own SANE for machines equipped with the MC68881 (which meets the IEEE standard) Apple had to provide the extra bit of accuracy it wanted in software, instead of utilizing the MC68881 to its fullest extent. This slows things down considerably.

Radius SANE, on the other hand, uses the MC68881 coprocessor to its fullest extent and does not attempt to provide any more accuracy than the IEEE standard. The result is a still very precise floating-point package but with the maximum possible speed.

The accuracy difference between Apple SANE and Radius SANE is minimal. Both packages use 80 bits for storing each floating point number. At this degree of accuracy, a floating point number can be specified to within approximately 2E-4932. Accuracy differences between the two SANE packages would be at this small order of magnitude.

WHAT ARE THE OTHER DIFFERENCES?

In order to provide maximum speed, Radius SANE does not implement the HALT mechanism of Apple SANE. The HALT mechanism permits a program to specify a halt handler routine for floating point exceptions. This routine will be called from within SANE whenever a floating point error occurs (such as overflow, underflow, etc.). The HALT mechanism was not implemented in Radius SANE since doing so would have slowed things down some and it is very seldom used in any case. A program can still set the halt handler but the routine will never be called from within Radius SANE. This does not prevent you from determining if an error occurred - you can still do so after the SANE call returns.

Another difference is that Radius SANE always rounds numbers to extended precision. Apple SANE lets you specify either single, double, or extended rounding precision. This feature was defined in the IEEE standard for simulating arithmetic systems on smaller machines. Unless you wish to simulate a computer that has only 32 or 64 bit floating point operations, there is no need for this feature. Again, this feature was left out because implementing it would have slowed down the coprocessor (the MC68881 works fastest when rounding to extended precision). You can try to set the precision to single or double, but your request will be ignored.

WHY DON'T ALL PROGRAMS RUN FASTER WITH RADIUS SANE?

Some programs may not run faster with Radius SANE for the simple reason that they are not using SANE calls to perform floating point arithmetic. In this case, they are using their own library of floating point routines that do not go through the Macintosh Toolbox interface. Microsoft Excel is a noted member of this class. If there is an MC68881 present, Excel will go directly to the co-processor to perform its arithmetic instead of going through the SANE interface.

Some development languages provide an option for compiling with inline floating point coprocessor instructions. Any program compiled with this option will not notice any speed improvement with Radius SANE since its floating point arithmetic will be handled directly by the coprocessor. The disadvantage to this is that the program will be unable to run on a machine without the coprocessor.

Another possibility is that a program might use only SANE calls that are not sped up by Radius SANE. These include operations on computational data types (64 bit integers) and decimal to string conversions. Radius SANE simply lets Apple SANE handle these operations since there would be little advantage in utilizing the MC68881.

The final possibility is that the program does not use any floating point arithmetic at all.

(June 9, 1988)