Subject: Re: IDL, arrays, and memory
Posted by Sean Raffuse on Tue, 04 Feb 2003 17:31:08 GMT
View Forum Message <> Reply to Message

> 
> However, whereas in C the difference in terms of speed and algorithm
> design is negligible between using a ragged array and a "wasteful"
> full 3D array, this is not true in IDL. In particular, you can't use
> most of IDL's fast array-based operators with an array of pointers;
> you're stuck accessing each element in a loop, which will be markedly
> slower for a data structure of this size.
> 
> You must balance the memory saved against the speed and flexibility
> with which you can operate on the data. This is a common theme in
> IDL, which, in many instances, trades increased memory usage for
> greater speed of execution. Often you can find other ways to organize
> the data which reduces the memory footprint while preserving much of
> the same flexibility had by putting it all in a single array. Or you
> can use, e.g., NaN values to fill the "wasted" array elements and
> avoid having to treat them specially.
> 
> Good luck,
> 
> JD

Exactly what I needed to hear. Speed of execution is indeed a major issue here. I'll stick with the big array.