Subject: Re: More efficient method of appending to arrays when using pointers?
Posted by Matt Francis on Tue, 04 Jan 2011 22:54:33 GMT
View Forum Message <> Reply to Message

On Jan 5, 9:44 am, nata <bernat.puigdomen...@gmail.com> wrote:
> Try this:
>
> temp = [TEMPORARY(*self.foo),next_array]
> ptr_free,self.foo
> self.free = ptr_new(temp,/NO_COPY)
>
> When you are creating the new pointer you are duplicating memory.
> The same occurs when you are retrieving the content of *self.foo. Use
> the function TEMPORARY and the keyword NO_COPY, your code will be more
> efficient.
>
> Cheers,
> nata

Thanks Nata. I see from the TEMPORARY docs that it works by using some scratch space IDL keeps allocated on hand. Do you know if there is an upper limit on the size of the arrays you use this approach with before you don't gain any efficiency (or start to be slower than not using TEMPORARY), for instance because you are using more memory than IDL keeps on hand for use with TEMPORARY?