Subject: Re: Read & write data files b/w IDL & Fortran 90
Posted by bridgemat on Wed, 14 Apr 2004 20:45:47 GMT
View Forum Message <> Reply to Message

Hey, thanks Paul! I'll give it a try. My actual data is all float/real
kind of stuff, so I hopefully wouldn't have this problem there. I was
just trying to see if I could do it with an "easy" case, but I guess
simple isn't always easy! :)

-Bridget

Paul Van Delst <paul.vandelst@noaa.gov> wrote in message
news:<c572r2tuc1@news.nems.noaa.gov>...
> Paul Van Delst wrote:
>> bridgemat wrote:
>>
>>> I did try the /f77_unformatted keyword when writing the original file,
>>> but I still couldn't get it to work. It shouldn't have anything to do
>>> w/ using Fortran 90 instead of 77, right? Or am I doing something
>>> altogether wrong in my Fortran program?
>>
>>
>> Hello,
>>
>> I just grabbed both your codes and mucked about and had a "d'oh" episode.
>>
>> You create the data in IDL using INDGEN == 2 byte ints. In your f90 code
>> you read the data with the default int which is 4 bytes. That's why
>> you're getting the I/O error.
>>
>> Two options:
>>
>> 1) In IDL create the data like so:
>> test=LINDGEN(3,4,5)
>> And read it with your current f90 code.
>>
>> Or
>>
>> 2) Create the data file with your current IDL code and in Fortran90,
>> define your integer arrays like so:
>> INTEGER, PARAMETER :: ip = SELECTED_INT_KIND(4)
>> INTEGER(ip), DIMENSION(3,4,5)::arr_in,arr_out
>>
>> The integer kind parameter from SELECTED_INT_KIND(4) should give you the
>> kind type for a 2-byte integer.
>>
>> I prefer option (1) coz it's the simplest, but YMMV.
>

> Apologies for my own followup, but in either option above you'll still need the
> /f77_unformatted keyword in the IDL routine.
>
> cheers,
>
> paulv
