Subject: Creating arrays of structures with Callable C
Posted by jicicuendez on Mon, 18 Apr 2005 17:29:23 GMT
View Forum Message <> Reply to Message

Hi everybody,

I am trying to use callable C/C++ code to generate a plot with IDL
from another application and there are a couple of things that I am
stuck with. Any help will be most appreciated.

First, in the IDL documentation, there is an example of how to create
and structure (page 176). This is done through a two-step procedure of
creating an IDL_STRUCT_TAG_DEF and then creating an structure with the
data itself such as

```
typedef struct time_series
{
   double x[100];
   double y[100];
   char RGB[3];
   int long nElements;
   IDL_STRING source[1];

}
TIME_SERIES;
```

The problem is that in principle the number of elements is not known
until is provided so I would like to substitute this by:

```
typedef struct time_series
{
   double *x;
   double *y;
   char RGB[3];
   int long nElements;
   IDL_STRING source[1];

}
TIME_SERIES;
```

and where memory has been allocated with new. Apparently the structure
can be created in the idl session with the IDL_MakeStruct function but
when I try to use such structure with and idl command it gives a
memory exception. Does anybody know how to avoid this problem?

Cheers,
Juan

Subject: Re: Creating arrays of structures with Callable C
Posted by Nigel Wade on Tue, 19 Apr 2005 09:09:03 GMT
View Forum Message <> Reply to Message

Juan I. Cicuendez wrote:

> Hi everybody,
>
> I am trying to use callable C/C++ code to generate a plot with IDL
> from another application and there are a couple of things that I am
> stuck with. Any help will be most appreciated.
>
> First, in the IDL documentation, there is an example of how to create
> and structure (page 176). This is done through a two-step procedure of
> creating an IDL_STRUCT_TAG_DEF and then creating an structure with the
> data itself such as
>
> typedef struct time_series
> {
> double x[100];
> double y[100];
> char RGB[3];
> int long nElements;
> IDL_STRING source[1];
>
> }
> TIME_SERIES;
>
> The problem is that in principle the number of elements is not known
> until is provided so I would like to substitute this by:
>
> typedef struct time_series
> {
> double *x;
> double *y;
> char RGB[3];
> int long nElements;
> IDL_STRING source[1];
>
> }
> TIME_SERIES;
>
> and where memory has been allocated with new. Apparently the structure
> can be created in the idl session with the IDL_MakeStruct function but
> when I try to use such structure with and idl command it gives a
> memory exception. Does anybody know how to avoid this problem?
>
> Cheers,
> Juan

The memory area used by IDL_ImportArray must be one, single, contiguous area of memory. So you can't do what you want in the way you have tried.

What you need to do is calculate exactly how much memory will be required by the structure, including the variable length arrays. You then allocate this memory using your favourite C allocation mechanism. Then you copy your data into this array. Finally you make the structure from this data using IDL_ImportArray. (NOTE: since you allocated the memory you are responsible for de-allocating it, so you need to provide a callback to do this).

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555